Anonymous Data Broadcasting by Misuse of Satellite ISPs

André Adelsbach, Ulrich Greveler and Sven Löschner
Horst Görtz Institute for IT Security
Ruhr-University Bochum
e-mail: {andre.adelsbach, ulrich.greveler, sven.loeschner}@rub.de

December 1, 2005

Abstract

Satellite ISPs connect users to the Internet by means of satellite communication. In this paper we discuss how to *misuse* satellite ISPs to allow any subscribed user to broadcast arbitrary content to a group of anonymous receivers. Exploiting the fact that the satellite downstream signal, containing the data requested by a user, is not only sent to this specific user only, but can be received in the whole footprint of the satellite we show how to broadcast certain data for an unlimited number of potential receivers. We conclude with open issues and future strands of work, such as sender anonymity.

1 Introduction

A satellite is a specialised wireless transmitter placed in terrestrial orbit for diverse purposes such as weather forecasting, television broadcast, radio communications, Internet access and GPS positioning. Satellites can receive and re-transmit thousands of signals simultaneously, from simple digital data to television programmes. Especially, in low-infrastructure areas they provide an interesting alternative, e.g., for high-speed access to the Internet, because they provide high data rates and cover very large areas with comparably low efforts.

The data packets in the downstream are broadcasted which makes it easy to receive the data of all users, not only the data packets for a specific user. Every person owning a DVB-S card and a digital enabled satellite dish is able to intercept all the data packets sent by the satellite. There are publicly available tools to watch data stream information in human readable form [7]. Moreover, interception of unsecured satellite signals for intelligence purposes is on the public agenda since the nineties [2].

Our Contribution In this paper we describe a way how an end-user can use (or *mis*-use) an satellite ISP to anonymously broadcast large amounts of data. The only pre-requisite for the sender is an ISP subscription and some necessary hardware (DVB-card, satellite dish). The anonymous receivers only need this hardware, there is no requirement for a subscription or Internet access at all.

Related Work The first proposal for anonymous (Internet) communication was published by David Chaum [3]. In his work so-called MIX servers are described that use layered encryption for cascading information via several servers in a way that an attacker cannot trace messages to a certain sender or receiver as long as he cannot control all the servers in the MIX network. Another approach for anonymous network communication is based on the peer-to-peer paradigm. A well-known system in this context is *Crowds* by Reiter and Rubin [9]. Here, http-requests are relayed via a chain of participating users' computers before being sent to the target web server. Responses are relayed backwards via the same chain to the anonymous user. *FreeHaven* [5] and *Freenet* [4] are distributed systems for anonymous and persistent data storage being robust against attempts to find and delete any stored data. *Tarzan* [6] is a fault-tolerant peer-to-peer anonymous IP network overlay being transparent to applications. Many other proposals regarding anoymous Internet communication can be found in [8].

2 Satellite ISPs

Satellite based ISPs come in two flavours:

- **One-Way:** In this lower cost variant, the satellite only handles the data downstream to the user with outbound data travelling through a telephone modem taking care of the low-bandwidth traffic from the user to the ISP. Most users only desire a high download bandwidth while they accept a rather small uplink capacity so this hybrid solution satisfies their needs.

- **Two-Way:** The more expensive two-way option lets the user have a satellite transmitter entity at their site that enables two-way communication with high bandwidth for up-link and down-link.[1] This option is more suitable for companies connecting their remote branches to a data network.

In this work we focus on the one-way variant, because it is more common for standard users today.[2] To illustrate how one-way satellite-based ISPs

[1] Note, that the up-link bandwidth is commonly still smaller than the down-link bandwidth.

[2] However, we want to stress that our proposal is even more suitable for two-way satellite communication, as the ISP has to broadcast all packets via the satellite down-link.

operate, consider the setting, where a user wants to download a MP3 file from some web server. A user establishes a small bandwidth dial-up Internet connection, e.g., an ISDN line, to the ISP. In order to initiate a download a request is sent through the dial-up line to an ISP proxy server, which relays the request to the desired destination. The reply coming from the server (e.g., the requested file) is re-routed by the satellite ISP so that it will not come back to the user's PC through the dial-up line. Instead it is encapsulated together with the user's specific IP address into a signal based on the DVB standard and the ISP ground station relays it to the satellite. The satellite broadcasts it back to the user who is running a piece of software on his PC which completes the TCP communication transparently to the application or operating system. Due to the broadcast character of satellite, the signal dedicated for this user can be received by anyone in the footprint of the satellite. In following section we describe how to (mis-)use a satellite ISP to broadcast to a large set of anonymous receivers.

3 Anonymous Data Broadcasts

Our basic idea is quite trivial, but very effective: we exploit the fact, that the satellite downstream, containing the data requested by the user, can be received in the whole footprint of the satellite. To broadcast certain data, e.g., a MP3 file, the sender first sends it to a dedicated server, which is connected to the Internet. Then the sender requests this data over the satellite ISP, which results in the data being broadcasted by the satellite ISP. The potential receivers simply listen to the satellite broadcast and filter the data, e.g., by implicit addresses.[3] Obviously, this system achieves unconditionally strong receiver anonymity due to the nature of a broadcast channel.

Our system works immediately if the satellite ISPs does not encrypt the broadcasted data. If the satellite ISP encrypts the satellite downstream using individual keys for each user, the system works as well, but is more involved: in this case the sender has to publish his session key, such that it is anonymously accessible by the receivers and enables receivers to decrypt the user's part of the satellite downstream. We cannot go into the details of this, because it strongly depends on the actual implementation of the ISP proxy software and would require illegal reverse engineering of this software. Fortunately, this effort is not necessary, as long as there are satellite ISPs which do not encrypt the satellite downstream (see e.g., [1]).

In the following we will discuss selected details of our proposal and how we implemented them in our Java prototype.

[3] An implicit address is an address, which allows nobody but the actual addressee to recognize that some message is addressed to him. Implicit addresses may be achieved by means of encryption of the broadcasted data, which, at the same time, achieves confidentiality of the broadcasted data.

Sender and Server The prototype of our sender first prepares the file the user wants to broadcast. Preparation involves splitting the file into chunks of constant size and encrypting each chunk. For encryption we use the Bouncy Castle crypto package [10], which offers a large variety of crypto algorithms. In particular, our prototype uses symmetric AES encryption, which may also serve as an invisible implicit address. For the first prototype we decided to add an explicit address, because it allows receivers to filter more efficiently (see below).[4] Key management is currently not implemented, i. e., we require that the sender and the receivers have exchanged a key beforehand.

Using standard HTTP(S), the sender uploads the encrypted chunks to the server (Fig. 1, step 1). For the upload functionality and sender GUI we extended the *Winie* network utility. This tool has been developed by W3C and is tailored to putting, getting and deleting files on a Jigsaw web server using the Jigsaw client side API [11]. The server is implemented on top of the W3C open source Jigsaw web server platform. We selected Jigsaw because it is lightweight, completely implemented in Java and has a modular architecture. The latter makes it very easy to extend the server's functionality: Using its `HeaderFilter`-class we can easily add the specific receiver ID to the HTTP-header.

After successful upload, indicated by a positive acknowledgement, the user can initiate the broadcast of his data. When the user clicks the *start broadcast*-button the sender software initiates the broadcast by sending a HTTP-request for his uploaded packets to its local proxy (Fig. 1, step 2), which forwards it to the ISP proxy via the dial-up connection (Fig. 1, step 3). The proxy of the ISP forwards the request to the server (Fig. 1, step 4). When the server receives the HTTP-request it answers by sending the requested packets to the ISP's proxy (Fig. 1, step 5). Now the satellite ISP forwards these packets to the satellite (Fig. 1, step 5), which broadcasts these packets encapsulated in a DVB stream (Fig. 1, step 7).

Receiver The receiver prototype uses the *jpcap* class package to capture IP packets from the DVB network interface, as provided by the tool *dvbnet*. Using this package the receiver software filters the IP packets being received on the DVB network interface by the IDs associated with the receiver. Captured packets are decrypted and temporarily stored. When all chunks have been received, they are joined again, which yields the complete file.

4 Conclusion and Future Directions

In this paper we proposed a practical way to achieve low-cost satellite broadcasts by *misusing* a satellite ISP. A first proof-of-concept prototype implementation was presented. Possible applications include file sharing, Internet radio or instant messaging.

[4]The price we have to pay for this is that broadcasts to the same group of receivers become linkable.

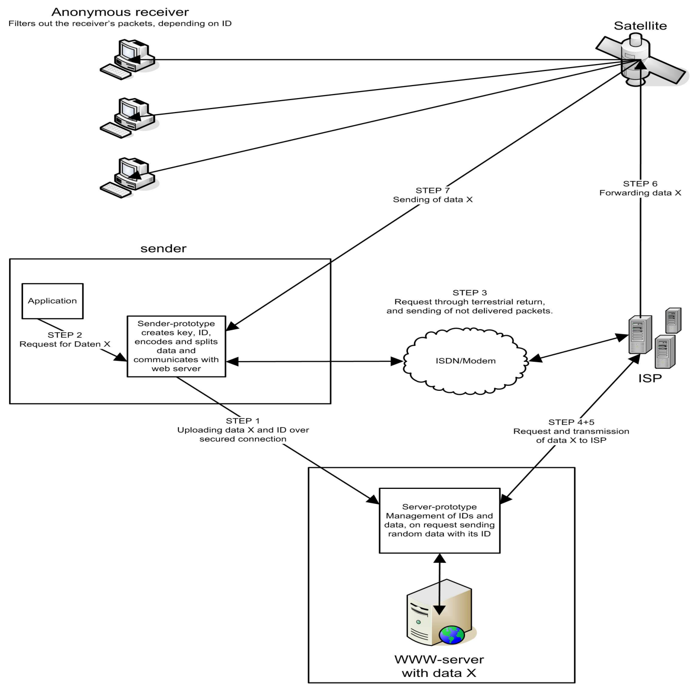

Figure 1: Anonymous data broadcasts via satellite

There are several open issues, which have to be addressed in future work. While unconditionally strong receiver anonymity follows trivially by the nature of a broadcast channel, achieving *sender anonymity* is more involved and requires a more advanced system design: one idea is to run a common server, where potential senders upload their encrypted data packets via some traditional point-to-point anonymizer. Now, instead of requesting its own packets, each sender requests *random* packets from the server, i. e., the party requesting a certain packet and, thereby initiating its broadcast, is different from the originator of this packet. This guarantees that nobody - not even the server - can tell who is the originator of a specific packet.[5] We consider this issue to be an important and challenging strand of future work. Another technical hurdle, requiring further attention is the high error rate of a broadcast downstream. Here, we need adequate redundancy to achieve robust broadcasts while causing minimal overhead. A related technical hurdle is that part of the requested data is returned via the low latency point-to-point dial-up (e.g., ISDN) connection, without being broadcasted. This problem may be solved by not acknowledging packets received over the dial-up connection, but requires further attention. For the future we pro-

[5]Obviously, the sender anonymity set only consists of those ISP customers requesting packets from this server and is significantly smaller than the receiver anonymity set.

pose an open-source project to continue the development of our prototype. If you are interested in the future development of this system feel free to contact us.

References

[1] Andre Adelsbach and Ulrich Greveler. Satellite communication without privacy – attacker's paradise. In Hannes Federrath, editor, *Sicherheit*, volume 62 of *LNI*. GI, 2005.

[2] Duncan Campbell. Interception capabilities 2000. Report to the Director General for Research PE 168.184, European Parliament, 1999.

[3] David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 4(2), February 1981.

[4] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. Freenet: A distributed anonymous information storage and retrieval system. In *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, pages 46–66, July 2000.

[5] Roger Dingledine, Michael J. Freedman, and David Molnar. The free haven project: Distributed anonymous storage service. In H. Federrath, editor, *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*. Springer-Verlag, LNCS 2009, July 2000.

[6] Michael J. Freedman and Robert Morris. Tarzan: A peer-to-peer anonymizing network layer. In *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS 2002)*, Washington, DC, November 2002.

[7] GNU General Public License. Dvbsnoop: a DVB / MPEG stream analyzer program. `http://dvbsnoop.sourceforge.net`.

[8] Free Haven Project. Anonymity bibliography. `http://www.freehaven.net/anonbib/bibtex.html`.

[9] Michael Reiter and Aviel Rubin. Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security*, 1(1), June 1998.

[10] The Legion of the Bouncy Castle. Bouncy castle crypto APIs. `http://www.bouncycastle.org/`.

[11] W3C. Jigsaw - W3C's Server. `http://www.w3.org/Jigsaw/`.