Introduction to Securing Data in Transit

Jennifer Vesperman

jenn@linuxchix.org

2002-02-24

Revision History

Revision 0.1 2002-02-17 Revised by: MEG
Converted from text file. Modified wording.
Revision 0.2 2002-02-23 Revised by: MEG
Incorporated Jenn's comments.
Revision 0.3 2002-02-24 Revised by: MEG
Conforming to LDP standards. Added abstract.

This article discusses ways to keep the transmission of data over the Internet private. In particular, authentication and encryption are covered.

1. Introduction

1.1. Copyright Information

1.2. Overview

The secure transmission of data in transit relies on both encryption and authentication - on both the hiding or concealment of the data itself, and on ensuring that the computers at each end are the computers they say they are.

2. Authentication

Authentication is a difficult task - computers have no way of knowing that they are 'the computer that sits next to the printer on the third floor' or 'the computer that runs the sales for www.dotcom.com'. And those are the matters which are important to humans - humans don't care if the computer is '10.10.10.10', which is what the computers know.

However, if the computer can trust the human to tell it which computer address to look for - either in the numeric or the name form - the computers can then verify that each other is, in fact, the computer at that address. It's similar to using the post office - we want to know if 100 Somewhere Street is where our friend Sally is, but the post office just wants to know where to send the parcel.

The simplest form of authentication is to exchange secret information the first time the two computers communicate and check it on each subsequent connection. Most exchanges between computers take place over a long period of time, in computer terms, so they tend to do this in a small way for the duration of each connection - as if you were checking, each time you spoke in a phone call, that the person you were talking to was still that person. (Sally, is that you? Yeah. Good, now I was telling you about the kids .. is that still you?)

It may sound paranoid, but this sort of verification system can inhibit what is called a 'man in the middle' attack - where a third party tries to 'catch' the connection and insert their own information. Of course, this relies on the first communication not being intercepted.

Public key encryption (see below) is the other common means of authentication. It doesn't authenticate the sender, but it does authenticate the receiver - and if both parties exchange public keys, and verify by some independant means that the key they have is the key of the party they wish to send to, it authenticates both.

3. Encryption

Encryption is the process of changing text so that it is no longer easy to read. A very simple example is the following sentence:

```
Guvf vf n fvzcyr fhofgvghgvba pvcure.
```

Commercial encryption uses methods which are a lot more secure than the one I used to produce that example. Almost all modern encryption methods rely on a key - a particular number or string of characters which are used to encrypt, decrypt, or both.

In the next sections, common encryption methods are presented. To illustrate how they work, ficticious characters named Bob and Alice will be introduced. Private key encryption and public key encryption are

discussed, as are their limitations.

3.1. Private key encryption

Private key encryption is the standard form. Both parties share an encryption key, and the encryption key is also the one used to decrypt the message. The difficulty is sharing the key before you start encrypting the message - how do you safely transmit it?

Many private key encryption methods use public key encryption to transmit the private key for each data transfer session.

If Bob and Alice want to use private key encryption to share a secret message, they would each use a copy of the same key. Bob writes his message to Alice and uses their shared private key to encrypt the message. The message is then sent to Alice. Alice uses her copy of the private key to decrypt the message. Private key encryption is like making copies of a key. Anyone with a copy can open the lock. In the case of Bob and Alice, their keys would be guarded closely because they can both encrypt and decrypt messages.

3.2. Public Key encryption

Public key encryption uses two keys - one to encrypt, and one to decrypt. The sender asks the receiver for the encryption key, encrypts the message, and sends the encrypted message to the receiver. Only the receiver can then decrypt the message - even the sender cannot read the encrypted message.

When Bob wants to share a secret with Alice using public key encryption, he first asks Alice for her public key. Next, Bob uses Alice's public key to encrypt the message. In public key encryption, only Alice's private key can unlock the message encrypted with her public key. Bob sends his message to Alice. Alice uses her private key to decrypt Bob's message.

The things that make public key encryption work is that Alice very closely guards her private key and freely distributes her public key. She knows that it will unlock any message encrypted with her public key.

3.3. Limitations of encryption

Cryptanalysis, or the process of attempting to read the encrypted message without the key, is very much easier with modern computers than it has ever been before. Modern computers are fast enough to allow for 'brute force' methods of cryptanalysis - or using every possible key in turn until the 'plain text' version of the message is found.

The longer the key, the longer it takes to use the 'brute force' method of cryptanalysis - but it also makes the process of encrypting and decrypting the message slower. Key length is very important to the security of the encryption method - but the 'safe' key length changes every time CPU manufacturers bring out a new processor.

Encryption does not make your data secure. Not using encryption, however, means that any data in transit is as easy to read as the contents of a postcard, sent in regular mail. Encryption at least ensures that anyone who does read your messages has worked hard at it.

4. Secure HTTP

Modern graphical browsers usually have a small 'key' or 'padlock' symbol at the bottom right or bottom left of the screen. When the 'lock' is closed or the 'key' is whole, the browser is encrypting the information and has exchanged basic authentication information with the other computer.

You can also check whether a browser is encrypting information by checking the URL at the top of the screen - any URL which uses 'https://' instead of 'http://' is using Secure HTTP. Unlike the locks and keys, the presence of 'https' does not mean that all your information is being sent encrypted, merely that it might be.

5. Secure Email

Most of the secure email programs use public key encryption. The receiver posts their encryption key somewhere public, somewhere that potential senders can locate it. The sender uses that key to encrypt the message, thus ensuring that only the receiver can decrypt it.

This works fairly well, but has the disadvantage that if your receiver isn't using a secure email program, or doesn't have a posted public key, you can't send encrypted mail to them.

Authentication is not a problem for secure email - provided the receiver has kept their private key secure, noone can easily decrypt the transmission.

6. Secure Shell

Networks often need remote system management, where the admin is in one building and the computer which needs attention is in another. Or programmers might work from home, using a remote access tool to use resources on the computers at work. The traditional tool which allowed these functions was a program called 'telnet', which gave access to a command line interface on the remote system.

The problem with telnet was that it sent everything using plain text. The modern version is SSH, which uses any of several encryption options and has a variety of ways to tell whether the user is authorised to connect to the host system. SSH stands for Secure SHell. "Shell" is a common term for the human/computer interface.

The SSH protocol has been used to provide secure ways to perform other common tasks. Different operating systems may have different tools, but common functions like copying or file management can often be done using tools which run over an encrypted SSH link.

SSH authenticates the computers on the first connection between each pair of computers - the two computers swap public keys, and on each subsequent connection check that the computers can decrypt a message. On the first connection, SSH depends on the user to verify that they have reached the correct computer. On subsequent connections, SSH will warn the user if it is uncertain of the remote computer's identity. Authentication security from that point is up to the user, though SSH will continue to encrypt traffic passing through it.

7. Other Data Transfer

FTP, IRC, ICQ and other systems are often used to transfer information around computers. If your program doesn't have an option to encrypt the data, assume that it is sent as plain text.

8. Final words

Secure data transfer methods are important, but the security can be broken without cryptanalysis if the computer at either end is insecure. Ensure that your computers and internal networks are secured.

Cryptanalysis can read most encryption methods, if the analyser is determined enough and wealthy enough to buy enough fancy hardware. If your information is worth enough, the most secure method of transfer might be 'sneaker-net' - a person with a floppy disk. If your information is that valuable, however, it's worth hiring a security expert to secure your networks. Take their advice.