Information hiding and covert channels

Václav Petříček
Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic.

Abstract. This paper maps the area of Information hiding and its subdisciplines steganography, watermarking, anonymity and covert channels. Examples of various techniques in information hiding are shown as well as some attacks and countermeasures against them. In the second part example of a covert channel via file access time is described and analyzed. Implementation using this covert channel under FreeBSD operating system is presented.

Introduction

As the name suggests information hiding is an art of hiding data and communication inside other data or communication.

Information hiding is somewhat similar to cryptography. On the other hand it seems to be stronger. Where cryptography hides the contents of communication, information hiding hides even the presence of the communication. In many cases hiding the contents of communication is not sufficient as the mere presence of communication, the quantity of messages and their size may disclose the contents of the communication. The method of analyzing the character of communication is called traffic analysis. Information hiding may be used to fight traffic analysis and has wide application also in military communication, espionage, wireless communication. Other use of information hiding is in copyright protection. This area is subject to intensive research that is motivated by the effort to publish music, books and videos online securely.

In contrast to cryptography, information hiding has not been as thoroughly studied yet. Many of the protocols and algorithms that are implemented in commercial products are proprietary and do not provide sufficient strength.

In this paper we will present an overview of the main subdisciplines of information hiding - steganograhy, watermarking, anonymity and covert channels. Than in the next part we will study an example of a covert channel in operating systea. This covert channel uses file access time that is used in many operating systems. We present an implementation using this covert channel.

Subdisciplines of Information Hiding

Information hiding may be divided into the following subdisciplines [1]:

- Steganography
- Watermarking
- Anonymity
- Covert channels

Steganography

Steganography tries to hide relatively large amount of information – it may be easy to remove but the most important thing is that it is unnoticeable. This subdiscipline of information hiding covers a variety of techniques from invisible inks, encoding of information using different colors for encoding information in first letters of words, sentences or slight modifying of the proportions or typeface of the letters.

In steganography, an encoded message is usually used for invisible point-to-point communication.

History of information hiding

Technical steganography The history of information hiding dates back to antiquity. Most of the early examples of information hiding demonstrate technical steganography – that is hiding information by technical means.

Histiæus for example tattooed a message, inciting a revolt against Persians, on the shaved head of his slave. It is interesting that this technique has been still used by German spies at the beginning of the 20th century. Later Sparta has been warned of Persian invasion by a message written under the wax on

a wax table. Messages have been hidden in the soles and earrings of messengers, encoded using different letter strokes or marked by punching small holes near letters in a cover-text. The last technique was still in use during the 17th century and further improved by employing invisible inks. The Germans also reused it during the 1st an 2nd world war.

Microdots hidden in ears, nostrils or under fingernails were used heavily during the 20th century. At that time invisible inks were deprecated by the invention of "universal developers", recognizing which parts of the paper had been wetted.

Interesting examples of steganography are various allegories like the *Vexierbild* painting by Shö, that looks like a landscape but when rotated by 90 degrees it shows portraits of famous kings.

Linguistic steganography Throughout the history, allegories were plentifully used to hide politically dangerous or inconvenient ideas. Hypnerotomachia Poliphili [9] is an anonymous work on forbidden love of a monk and a woman. First letters of the chapters give a sentence that can be translated as "Brother Francesco Colonna passionately loves Polia." Colonna was still alive when the book was published.

One of the most famous acrostics is *Amorosa visione* by Giovanni Boccaccio. The first letters of the succeeding tercets create three other sonnets.

Monks used word tables that produced texts looking like prayers and spells. They also used musical notes and geometrical drawings to hide secret messages.

Paper masks have been invented in ancient China to hide messages in random parts of a cover-text.

Classification of Steganography

Prisoner's problem Simmons [4] first introduced a model for invisible communication. Let's assume that Alice and Bob are in prison and the only way they can communicate is through a warden named Wendy. Wendy will check all messages passing between them and will not let through anything that looks suspicious or like ciphertext. Alice and Bob have to establish a subliminal channel by exchanging innocent messages with hidden meaning. In addition to passive monitoring Wendy may act as an *active warden* and alter the messages exchanged between Alice and Bob. Moreover she may even forge messages pretending to be the other prisoner and act as a *malicious warden*.

Steganography deals with the situation in the following way (see Figure 1). Alice uses a key to insert a message in a cover object and creates a stego-object. The stego-object can be sent to Bob. Bob applies his key to extract the original message.

It is necessary to be aware of the danger of choosing the same cover to embed different messages. Wendy would notice the differences and find out about the subliminal channel. The stego object also has to look like an ordinary and innocent message – random data will be filtered by Wendy.

If Wendy acts as an *active* or *malicious* warden, all the messages transmitted have to be authenticated.

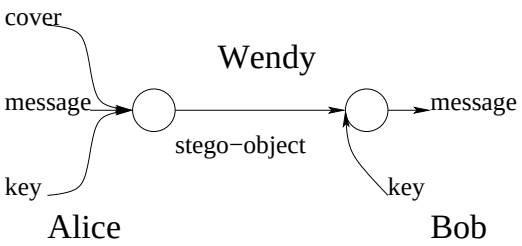

Figure 1. Schematic description of steganography

Pure steganography Security of this technique is based on the knowledge of the way information is inserted in the cover – a concept known as "security through obscurity". It is used to hide information into audio, video, images and text. It tries to preserve the highest similarity of the cover and stego object possible so that the additional information is not noticeable by a casual viewer. This may be achieved by choosing the best suitable cover (eg. by scanning the image repeatedly until the the most suitable cover is found).

Characteristics of inserted data can be hidden by encrypting it prior to insertion. Compression also reduces the redundancy of the inserted data and the amount of changes necessary for the embedding.

Secret key steganography Secret key steganography is similar to secret key cryptosystems. Inserted message is encrypted by a secret key that also controls the location where the data will be hidden

in the cover. Original message can be recovered by using the same key.

Some techniques derive the key directly from the cover and thus try to avoid the key distribution problem. This is how the secret key steganography is being reduced to pure steganography.

Public key steganography Public key steganography is analogous to public key cryptosystems. The message is encrypted by using the public key of the receiver and inserted into the cover instead of the original "noise". Security of this method is based on the assumption that it is not possible to distinguish the ciphertext from "natural noise". It is also susceptible to the "Man in the middle attack".

Steganalysis Steganalysis – a means of attacks on steganographic systems exploits the statistical deviations and characteristic marks that leave behind different steganographic tools.

Similar to one time pad – that is an unconditionally secure cipher – a perfectly secure steganographic system exists. The problem is that it requires transmission of random data and is therefore useless.

Examples of steganography According to Westfeld [10], it is possible to store a GSM channel (8Kbps) inside an ISDN video conference channel (64Kbps) without severely degrading quality of the video signal.

Other techniques have been implemented in various tools to embed information in audio or images. These techniques often use the LSB substitution or in the case of images palette manipulation.

Anderson [11] proposed a steganographic file system that can store files accessible only with a correct name + key pair. Without the knowledge of the key, the presence of any file cannot be verified.

Watermarking

Watermarking differs from steganography in the fact that it also requires the watermark to be robust. Requirement of robustness means that it is infeasible for an attacker to remove the watermark from the carrier.

Watermarks embedded in LSB of the cover are easy to remove by compression or filtering. It is therefore necessary to embed it in more significant parts of the cover. *Supraliminal channels* are created this way. When judging the robustness of a watermarking method it has to be specified what set of transformations we consider. Some watermarks may be highly resistant to jpeg compression but not to other transformations.

History of watermarking The name of this technique has been inspired by the 13th century paper watermarks that were used to distinguish the quality and manufacturer of the paper. Today, special watermarks are used in bank notes to prevent photocopying. Watermarks embedded in images are used to track illegally published copies on the Internet.

Classification of watermarking methods

Visible x invisible watermarks Watermarks may be visible or invisible. Visible watermarks can be detected by bare eyes. Visible watermarks may be used for printed documents authentication, copyright protection and traitor tracking. Invisible watermarks are more suitable for digital publishing as they do not degrade the quality of the product, but they are usually less robust.

Fingerprinting Fingerprinting is considered a subclass of watermarking. It is a technique designed for traitor tracking. This is achieved by "fingerprinting" every copy of data with a unique ID that is used to identify the customer that purchase it. This way every illegal copy can be traced back to the traitor, who distributed the illegal copy. Fingerprinting schemes have to be resistant to collusion attacks where reasonably sized groups of customers cooperate, trying to remove or forge the fingerprint.

Copyright protection Copyright protection is the most prominent application of watermarking today. With its boom, recording industry wants to use the Internet as a distribution media but is afraid of the ease with which the digital content can be copied.

By embedding a watermark the copyright of a work may be retained and illegal copies may be identified. Executable code can be compiled in such a way that the author's copyright is encoded by arranging the order of instructions. Similarly, the order of words and the phrases used in a text may identify the copyright holder.

Copy protection Copy protection may be achieved only in closed proprietary systems. There the player may refuse to play a song or movie that has not been purchased for replay on this equipment.

Attacks There are various means of attacks against watermarking systems. The watermark has to be robust against collusion attacks, multiple watermarking by different parties and removal of the watermark by transformations that do not degrade the usability of the digital object.

Anonymity

Anonymity is in fact identity hiding. Different protocols provide sender anonymity, receiver anonymity or both of them.

Anonymity is one of the most controversial issues. But even though it may be abused, there are cases when staying anonymous is desirable and perfectly acceptable. For example location anonymity is especially important in distributed systems seeking to provide high data availability.

Classification of anonymity systems Depending on the degree of anonymity they provide, the systems may be classified as follows.

Sender anonymity Systems providing only sender anonymity are represented by anonymous remailers, the Crowds [12] and Anonymizer [13]. They are used where the initiator needs to stay unrecognized.

Receiver anonymity Receiver anonymity is provided by various anonymous accounts (nyms). Thus it is possible to be contacted through a trusted third party that will not reveal the real identity. To achieve a higher degree of security this approach may be chained using multiple trusted third parties and every trusted party in the chain has to be compromised. Goldberg [5] proposed a technique for anonymous publication through a *rewebber network*.

Sender and receiver anonymity Onion routing is a combination of the former two techniques and provides both communicating parties with anonymity. Eternity Service [6,16] is using onion routing to protect the location of stored data in it as well as that of the clients. Remailers may also provide sender and receiver anonymity.

Attacks Traffic analysis tries to reveal identities by watching who is communicating with whom. Tracking of anonymous surfers based upon their preferences and browsing habits is used for marketing purposes.

Covert channels

Unintentional communication paths that were not primarily designed in the systems and protocols for communication purposes, but can be used that way, are called *covert channels*. Some fields in communications protocols can be used for passing information even though their primary purpose is different. One example of such a field is the initial sequence number in TCP protocol. Other examples of covert channels are image downgrading [2] and digital signature schemes [14]. They are usually used for "leaking" information. Covert channels in operating systems allow processes to communicate "invisibly" and possibly across different security zones specified by security policy. CPU load, filenames of temporary files and in fact any shared resource can be used for covert communication.

Classification of covert channels

Noisy x noisless channels Covert channels may be noiseless – in case the bits transmitted are read correctly with probability of 1. Otherwise the channels are noisy. Channel capacity can be derived by using Shannon's theory [8]. On a noisy channel it is necessary to utilize some kind of error correction or detection mechanism.

Storage x timing channels Channels are also classified according to the media they use for the transmission. Storage channels are those that use some direct or indirect writing to a storage medium for signaling. Timing channels are based on the fact that processes access a shared resource like CPU. The response times and delays in getting access to the resource may be used to transmit the information.

Aggregate x nonaggregate channels Other type of classification of covert channels relates to the degree of their aggregation. Multiple nonaggregate channels may be combined to get an aggregate channel. The speed of transmission can be thus improved considerably.

Countermeasures against covert channels It is very difficult to prevent covert channels and in practice sometimes impossible. The system has to be subject to syntactic flow analysis. Covert channel analysis, documentation and monitoring has to be done to achieve B2-A1 certification [7].

In the next section we will deal with a covert channel via file access time.

Using file access time as a covert channel

It may seem that denying processes of all the traditional ways of communication like sockets, file sharing IPC, etc., is sufficient to fully separate them. We will see that this is not true and that it may still be possible to communicate.

Let's assume that the processes cannot communicate directly, but have read-only access to some system-wide configuration files and directories. Then they are able to utilize the file access time for covert communication.

The original purpose of file access time is to help the system administrator with finding unused files, but we may use it in the following way. When a process reads the file during a specified period, it means a 1. When it is not read, a 0 is assumed.

Throughput

As file access time is stored in seconds, we can easily signal 1bps. It is possible to signal more bits per second by reading the file and checking the access time more frequently, but we have to expect higher error rate, depending on the load of the machine. The second possibility for improving throughput is to aggregate more channels together. These two approaches may also be combined.

Error control

Errors in communication may occur when other processes access the files used by our covert channel. As these errors are one way (0's changes to 1's) but random (see Table 1), it is not effective to use cyclic or "m of n" codes.

Error detection x error correction Error detection codes are used in ARQ (Automatic Replay Request) techniques. If the receiver detects and error she has to signal it back and wait for retransmission. On the other hand, FEC (Forward Error Correction) relies on the redundancy of the code to repair all errors. We chose to use error detection due to its lower overhead and the fact that in this case it is not necessary to signal errors back, as the sender can see if an error occured.

Implementation

We implemented `covertalk` application that resembles unix talk program and has similar functionality. The difference is that `covertalk` communicates over a covert channel via file access time. The source code may be downloaded from [15].

Platform `Covertalk` has been implemented on FreeBSD 4.3-RELEASE operating system and coded in C using the ncurses library.

Choice of files We chose to use the top level directories and files for the communication because:

- They are rarely accessed - most accesses being done by nightly periodic scripts doing system maintenance.
- They rarely change. The top level directories are always present and hardly ever moved or deleted. By contrast, the files in temporary directories change frequently.
- All users usually have read permissions to top level directories and files.

Table 1. Number of accesses to some top level files and directories.

quiet machine[2]		busy machine[3]	
file	# of accesses[1]	file	# of accesses[1]
/usr	8	/etc	36
/sbin	8	/bin	14
/bin	8	/sbin	14
/etc	7	/usr	12
/home	5	/home	6
/tmp	4	/tmp	4
/var	2	/var	2
/root	2	/root	2
/modules	2	/modules	2
/mnt	2	/mnt	2
/dist	2	/dist	2
/dev	2	/dev	2
/boot	2	/boot	2
/lost+found	2	/lost+found	2
/kernel	0	/kernel	0
/COPYRIGHT	0	/COPYRIGHT	0

[1]The numbers are averages per 24 hours obtainetd over a period of 5 days.

[2]Single user desktop machine (FreeBSD 4.3-RELEASE) running XWindows.

[3]Mail server (FreeBSD 4.3-RELEASE) with shell accounts and about 100 user logons per day.

Source coding and error correction As the talk application requires transmission of mostly printable ascii characters, we can encode it in 7bits and use one additional bit for error detection. due to the low error rate (see table 1), this encoding together with even parity is sufficient.

To achieve the throughput of 1 character per second, we need at least 8 files for transmission and the same number for receiving. no additional files are needed as the control characters can be sent inline.

Synchronization To synchronize the sender and receiver, every one of them starts by sending a sequence derived from passphrase entered upon startup using a one-way hash function. when the other side detects this "carrier" it starts sending the text entered by the user.

Conclusions

We have presented a brief overview of current research areas of information hiding: steganography, watermarking, anonymity and covert channels. especially watermarking is subject to intensive research these days, as it has direct applications in copyright protection on the internet.

We also presented a sample application that uses covert channel via file access time. even though it is quite easy to disable this covert channel by allowing only owner of the file to read its access time, many other covert channels using other share resources exist. That is why securing removing covert channels from current operating systems seems infeasible.

References

[1] Katzenbeisser, Stefan, Fabien A.P. Petitcolas: information hiding techniques for steganography and digital watermarking, *artech house*, 1999.

[2] Kurak, Charles, John McHugh, a cautionary note on image downgrading, *computer security applications conference*, 1992, pp. 153–159.

[3] Newman, B.: secrets of german espionage, *robert hale ltd.*, 1940.

[4] Simmons, G.J.: the prisoners' problem and the subliminal channel, in advances in cryptology, proceedings of crypto '83, *plenum press*, pp. 51–67, 1984.

[5] Goldberg, I., and D. Wagner: TAZ servers and the rewebber network: enabling anonymous publishing on the world wide web, *first monday* , vol. 3 no. 4, 1998.

[6] Anderson, R. J.: The Eternity Service, *in proceedings of pragocrypt '96*, 1996.

[7] A guide to understanding covert channel analysis of trusted systems – light pink book, *national computer security center – rainbow series*, 1993.

[8] C. E. Shannon: Communication theory of secrecy systems. *bell system technical journal*, pp. 656–715, 1949.

[9] Anonymous: hypnerotomachia poliphili: The dream battles of Polia's lover, 1499.

[10] Westfeld, A., Wolf, G.: Steganography in video conferencing system, *in proceedings of the second international workshop on information hiding,vol. 1525 in lncs, springer*, pp. 32–47, 1998.

[11] Ross Anderson, Roger Needham and Adi Shamir: The steganographic file system. *in proc. second international information hiding workshop*, 1998.

[12] Reiter, Michael K., Aviel D. Rubin: crowds: Anonymity for web transactions. *dimacs technical report*, 97(15), 1997.

[13] Cottrell, L. Anonymizer.com. web site at `http://www.anonymizer.com/`, 1997.

[14] Simmons, g.j.: The subliminal channel and digital signatures, *in advances in cryptology, proceedings of eurocrypt '84, vol. 209 of lncs, springer*, pp. 364–378, 1985.

[15] Petricek, V.: `http://www.kolej.mff.cuni.cz/~petricek/work/covertalk/`.

[16] Benes, T.: The Strong Eternity Service *in proceedings of the fourth international workshop on information hiding, lncs, springer*, 2001.