Parasitic Communications

Covert channels of communication hidden inside legitimate networks cannot be eliminated but they can be significantly reduced by careful design and analysis.

BY CARLOS SCOTT AND CHEZ CIECHANOWICZ

Royal Holloway University of London

INTRODUCTION

In any modern organisation, leakage of confidential information ranks among the highest fears of any executive. Currently, the most common means of information leakage are employees discussing proprietary information outside their employment context, reproduction of hard copies of classified documents and copying of confidential information on portable media. However, most organisations have taken measures to prevent such leakage, which have led to an increase in computer-based data smuggling.

As most organisations depend on broad and heterogeneous communication networks, someone could smuggle out sensitive private information in a number of ways, and detecting this can be a challenge. There are plenty of tools that can help you inspect outbound e-mail, web traffic and other forms of network communications, but it's no easy task identifying which information is leaving the organisation legitimately or not. It may even be impossible to determine if a communication is occurring at all, as network communication channels can be abused to implement covert communication channels.

Network Covert Channels– Parasitic Communications

Carlos Scott
Security Monitoring Team,
PSA – Service Management,
Vodafone UK
Newbury, U.K.

Chez Ciechanowicz
Information Security Group,
Royal Holloway, University of London
Egham, Surrey, U.K.

This article was prepared by students and staff involved with the award-winning M.Sc. in Information Security offered by the Information Security Group at Royal Holloway, University of London. The student was judged to have produced an outstanding M.Sc. thesis on a business-related topic. The full thesis is available as a technical report on the Royal Holloway website http://www.ma.rhul.ac.uk/tech.

For more information about the Information Security Group at Royal Holloway or on the M.Sc. in Information Security, please visit http://www.isg.rhul.ac.uk.

COVERT CHANNELS? WHAT ARE THEY?

A covert channel is a communication channel that is not designed and/nor intended to exist, and that can be used to transfer information in a manner that violates the existing security policy.

The existence of covert channels comes as a result of protocol specifications often being vulnerable to 'misuse' in unintended or unanticipated ways. This is due mainly to language peculiarities, lack of use of formal methods to define them, and the complexity of expressing the concepts that form the base of a protocol in an unequivocal manner. They often include extendable and optional

elements, which are normally not disallowed explicitly in situations where they have no use. Differences in implementations are also normally allowed, which reduces consistency. Covert channels adhering to protocol specifications may be established in these cases and their detection is extremely difficult, as the resulting traffic cannot be considered anomalous.

MALICIOUS USE

Other than data exfiltration, covert channels can be used for a variety of purposes, depending on the goals of the malicious user. Some of these pose serious threats:

- Criminals can use them for covert communications. The use of encryption provides confidentiality to communications channels, but it does not prevent the detection of communication patterns, which is often sufficient to discover the onset of anomalous activities. This makes covert channels, which are harder to detect, particularly useful in information warfare scenarios.
- Hackers that have compromised systems normally use them as launching points for subsequent attacks. Covert channels can be used to send instructions to these systems; if this traffic is not covert it would alert the systems administrator, who would easily discover the compromised systems. They can also be used as backdoors, as the intruders cannot rely on the initial exploitation vector remaining available.[1]
- They can be used to circumvent measures taken by governments and private organizations to limit the freedom of speech and civilian use of strong encryption.

A good indication of the threat they pose can be derived from the highly publicized Distributed Denial of Service (DDoS) attacks conducted against popular internet websites such as Yahoo!, CNN, E-bay, E-trade and Buy.com in 1999. These were automated attacks using thousands of distributed agents, which communicated with each other through covert channels in network protocols. Another good example that exposes the risk is the suspected hidden transmission of plans or instructions through the Internet to terrorist groups operating within the United States. It is believed that many of these messages were transmitted using covert channels, encrypted and embedded within innocent-looking files.[2]

The main objective of a covert channel is to hide the fact that a communication is taking

The main objective of a covert channel is to hide the fact that a communication is taking place.

place. Cryptography is different, which does not conceal the communication, but rather makes the information being communicated available solely to the intended receiver.

STEGANOGRAPHY

Covert channels and steganography (Greek for covered writing) are closely related and often confused. Although they both involve data-hiding and data transmission, the techniques differ significantly. Examples of steganography are the manipulation of bits in image or audio files to conceal information. A variety of things can act as a conduit for steganographic communication. A Greek historian tells that a messenger's head was shaved and tattooed with a secret message calling for revolt against the Persians. Later, the messenger travelled to the location of the intended receiver after the hair had re-grown. The head was shaved again and the message revealed to the receiver[3]. While steganography requires some form of content as cover, network covert channels require a network protocol as a carrier. The ubiquitous presence of some network protocols (e.g. the Internet Protocol) makes covert channels highly available and usable even in situations where steganography cannot be applied.

As network covert channels are communication channels that are neither designed nor intended to exist, the communication streams must be embedded inside authorized channels. They may be based on existing protocols from OSI low layers (e.g.: IP, TCP, UDP) to OSI high layers (e.g.: HTTP, SMTP). However, the carrier protocol must be authorized by the Network Access Control System (NACS), and require some trade-off between reliability and covertness.

It is often possible to use more than one of these channels, and even different types of these channels, as simultaneous carriers for the covert information. Each one of these channels has its own requirements in terms of quality attributes. For example, different communication channels over HTTP, ICMP and SMTP protocols may be used simultaneously with the objective of increasing the stealthiness of a control channel. However, the use of multiple communication channels might be counter-productive for the attacker in terms of stealthiness, as it may alert NACS administrators of the existence of ongoing covert communications. Another possibility is the aggregation of multiple data and/or control channels over a single communica-

Examples of steganography are the manipulation of bits in image or audio files to conceal information.

tion channel. This is most useful when the NACS in place has a very restricted number of permitted communication channels. As this approach involves multiplexing several channels over a single one, bandwidth usage of the latter may increase significantly, which may lead to the detection of the covert communications.

COVERT CHANNELS ON COMMON NETWORK PROTOCOLS

The Internet Protocol version 4 (IPv4) header tunnelling was one of the first instances of covert channels on the network layer. Since it is an enabler for wide area networking, which means that its scope is global and can cover disparate network subsystems, IP is a very popular target protocol for data hiding. Each packet contains a protocol header that consists of 23 fields used for a variety of purposes, such as the carrying of routing information, Quality of Service information, and fragmentation. However, this variety results in an inherent risk of them being used to transmit data instead of network management information. Data can be transferred covertly between networks, by compressing data to a form that can be embedded in the header. Although these headers are open to inspection, the embedded value is considered legitimate and is not considered anomalous.

The 16 bit IP Identification field is the most eligible choice. It can be used for byte-to-byte covert communication. The protocol specification states that it is used to identify individual packets when packet fragmentation occurs in the network. A covert channel can be created by encoding data in separate 16 bit values and transmitting them in the IP ID field, then decoding them at the other end.

Many other similar vulnerabilities exist in the IP protocol, involving the manipulation of IP header fields, such as the 24-bit options field, the 8-bit padding field, the 3-bit Don't Fragment (DF) flag and the Time to Live (TTL) field.

A new enhanced version of the Internet Protocol, known as the Internet Protocol version 6 (IPv6), is intended to replace IPv4 in the coming years. It provides improved reliability, much larger address space and better security than its predecessor. But it is also much more complex, making it more vulnerable to being used for covert communications.

Another commonly used protocol normally targeted for covert communications is the TCP protocol, designed for the provision of

Data can be transferred covertly between networks, by compressing data to a form that can be embedded in the header.

packet reordering on arrival at the receiver and the provision of a retransmission service that allows the receiver to request the retransmission of particular segments. Initially, an Initial Sequence Number (ISN) is generated randomly, which is used in the first segment of a TCP session (SYN segment). However, the use of a non-random value in the sequence number field doesn't disrupt the TCP protocol. This implies that a malicious user can use this field to transmit 32 bits of arbitrary data per segment. Furthermore, because random values are normally expected in this field, covert channels using it as a carrier are particularly hard to detect.

Going up the protocol stack, the application layer offers endless possibilities for establishing covert communications. Almost all organizations allow the use of the HTTP protocol, as the World Wide Web is the primary information resource. It is universally implemented and used across different types of networks, which makes it an interesting target for carrying covert channels. Lower layer protocols (IP, TCP, ICMP) present numerous limitations, such as limited capacity and modification of packets at intermediate nodes. For this reason HTTP has become the most frequently used protocol for covert communications, and several different mechanisms for covert data transfer have been proposed and implemented.

HTTP request messages may contain multiple headers, some common examples being *User-agent and Referer*. Malicious users can exploit this by using headers to transmit arbitrary data. A particularly interesting feature of the HTTP protocol is the Entity-body. It is normally only used in HTTP POST requests, because it has no real use for other types of requests. However, the protocol specification doesn't state explicitly that this should not be present in other request types, which enables the transmission of arbitrary data by a malicious user in any request type. Just like HTTP request messages, HTTP response messages may be exploited in a similar fashion.

Recently, covert channels for transferring information through various VoIP protocol specifications such as control traffic (i.e. SIP, H.323, RTCP) or data transport protocols (i.e. RTP) have been discovered. They present a significant risk; emerging threats such as VoIP spam or botnets may work in tandem to transfer control signals or binary executables through VoIP covert channels.

A proof of concept attack demonstrating

Going up the protocol stack, the application layer offers endless possibilities for establishing covert communications.

this new VoIP threat has been developed (Vo2IP). It allows for the establishment of a hidden conversation by embedding further compressed voice data into regular PCM-based voice traffic (i.e. G.711 codec). Therefore an eavesdropper who is not aware of the use of the covert channel can't decode the conversation properly.[4]

X.509 DIGITAL CERTIFICATES AS CARRIERS OF SECRET DATA

The modification of any value in a digital certificate can be detected by computing the digital signature of the certificate with the algorithm specified and comparing it to the signature, which must be decrypted using the Certificate Authority's (CA) public key first. Thus, the use of certificates to exfiltrate data might seem like an impossible task. However, an attacker trying to exfiltrate data from a private system is not really concerned with verifying the validity of the key contained in the certificate, as he/she is using it for other purposes, so he/she might accomplish the objective even without completing the signature verification process. In some extreme cases the data could even be encoded in the signature or the public key.

It is evident that this approach is a very naïve one, as legitimate users of the system will detect that the certificate has been modified when they try to verify the certificate's signature. This means an attacker has to find a way of sending modified certificates only to his client(s), and the correct certificate to legitimate clients. If the attacker is trying to exfiltrate data, it is reasonable to assume that he has some control over the private system. He could choose to send the secret data embedded in the certificate only in certain cases, based on the source or destination address of the underlying packets or on the occurrence of particular events controlled by him (i.e. specially crafted requests to a server).

A network-based detection system, which can be even based on simple traffic analysis, could detect suspicious values in the different certificate fields used by the attacker (i.e. strange serial numbers). This means the attacker has to focus on encoding the data in a way that makes it look similar to that of a genuine certificate. A network based detection system based on more complex techniques can execute the signature verification process and discover that the certificate has been modified, although this would require that the detection system perform

A network-based detection system, which can be even based on simple traffic analysis, could detect suspicious values in the different certificate fields used by the attacker (i.e. strange serial numbers).

this for every certificate request, which can prove costly in terms of computing resources.

Self-signed certificates constitute a particular cause for concern. The signature on a self-signed certificate is generated with the private key associated with the certificate's subject public key, which proves that the issuer, who in many situations is the user, possesses both the public and private keys. Their use presents a much more interesting scenario for the malicious user trying to smuggle data. There is a possibility that the compromise of the private system might lead to the compromise of its private key, which is infeasible in scenarios where trusted third parties (TTPs) are used to sign the certificates, since compromise of the TTP's private key is extremely unlikely. In cases where the detection systems only verify the signature of the certificate, and do not perform analysis on the certificate itself, the transmission of secret data within the certificate can prove undetectable. The use of self-signed certificates is not uncommon, in particular in Public Key Infrastructures (PKI) based on the Web of Trust scheme. One could argue that if the attacker has compromised the system to the extent of compromising the user's private key and the ability to generate a digital certificate with secret data, he could find an easier way to exfiltrate data (rather than having to embed it in a public-key certificate). But it is important to remember that the objective of using covert channels is not to hide the data being exfiltrated, but to hide the fact that the transmission is actually taking place, thus making it appear as regular traffic.

A particularly interesting way of achieving this could be to manipulate the validity dates of a certificate so that the secret messages are represented as apparently valid time spans. The certificate validity period is the time interval during which the CA guarantees that the certificate information is accurate. The field consists of a sequence of two dates: the date on which the validity period begins (notBefore) and the date which marks the expiry of the certificate (notAfter).

A covert channel can be established by encoding data as the difference between these two values. According to the X.509 specification, the minimum year value allowed for the notBefore date would be 1900. The notAfter data could have a theoretical maximum year value of 9999. Both notBefore and notAfter time values must be

It is important to remember that the objective of using covert channels is not to hide the data being exfiltrated, but to hide the fact that the transmission is actually taking place, thus making it appear as regular traffic.

specified to a precision of seconds. This means that the values allow for the following number of differences, which is equivalent to the number of seconds that exist between 00:00:00 of the year 1900 to 23:59:59 of the year 9999:

(9999 – 1900) * 365 * 3600 = 10, 642, 086, 000.

If a variation of one second is used to represent a message or symbol, this means that **10.642.086.000** symbols could be encoded using this differential encoding scheme, roughly equivalent to 233 messages. Thus, up to 33 bits of data can be transferred covertly using this time-differential encoding scheme in a single digital certificate by manipulating its time validity.

A value of the notAfter field earlier than the current date and time would be highly suspicious, as it would indicate that the certificate has expired. That would reduce the amount of possible symbols that can be encoded using this scheme. Furthermore, the use of border values allowed for the validity of a certificate might also raise suspicion, as their use in legitimate situations is questionable (it would generate in practical terms a certificate that never expires, or a certificate whose use was permitted even before the invention of modern computer systems). The range should be chosen carefully in order to look innocuous, taking into consideration the current date and the standard validity periods used in different scenarios. Certificates with expiry dates of three or more years from the issuing date are not uncommon on the Internet, although those used in mid to high security systems can have much shorter validity periods (a few days, hours or even minutes).

An interesting scenario would be to take advantage of different fields in the *Transport Layer Security* (TLS) protocol; these have been identified as potential carriers for covert channels to request the exfiltration of data from a compromised server[5]. As mentioned earlier, an attacker trying to exfiltrate data in X.509 certificates might need to send modified certificates only to client(s) under his control, while still being able to send the appropriate certificate to legitimate clients. This might be accomplished by sending instructions through data encoded and embedded in the aforementioned TLS fields. There must be processes on the server capable of interpreting and executing such instructions.

Other useful attributes that can be specified by the attacker are the types of encoding that should be used, e.g. encrypt the data using a specific algorithm.

These covert channels in the TLS protocol could also prove useful in establishing different parameters to be used when exfiltrating data via X.509 certificates. For example, the attacker could inform a process (on the compromised system) about the location of the required data, i.e. in which specific fields and positions of the X.509 certificate to embed the data. Other useful attributes that can be specified by the attacker are the types of encoding that should be used, e.g. encrypt the data using a specific algorithm. If the malicious user is employing the validity fields of the X.509 certificate (as explained above), he could use the TLS covert channel to set the notBefore or notAfter values to be used, chosen according to the context.

UNCOVERING THE SECRET

Covert channels cannot be completely eliminated, although they can be considerably reduced by careful design and analysis. A portion of the bandwidth of a legitimate communication channel that can be sidetracked, to be used as a carrier for covert communications, will always be present.

The assessment of covert channels through the use of probabilistic risk management is very complex. Security management standards, such as ISO 17799 and ISO 27001 do not treat covert channels explicitly, but assume they are managed with broad network segregation and network connection controls. A framework that allows for the holistic identification of network covert channels is yet to be developed.

The detection of covert channels can be approached in a number of ways. One such technique consists of the monitoring and detection of traffic that exceeds specific thresholds established previously at the network and/or transport layers. A signature-based detection approach is also valid, in which case the traffic is monitored for the occurrence of characteristic patterns that occur during the establishment of covert channels. The detection of protocol anomalies generated by some tools is also an indicator of the presence of covert communications. Another approach is to learn the "network behaviour" and to use statistical methods to establish if the monitored network traffic is "behaving correctly".

Standard IDS technologies are also commonly used to aid in the detection of covert communications, along with the deployment of *Network Security Monitoring* (NSM) models.

Covert channels cannot be completely eliminated, although they can be considerably reduced by careful design and analysis.

It is critical for detection techniques to limit both the number of false positives and the number of false negatives.

A covert communication channel might remain hidden if the detection systems in place are signature-based and there is no specific rule that will find the channel. This is also true if the set-up of such a rule is too costly, either in terms of system resources, money or false-positives. Furthermore, it is possible to increase the difficulty of detection of covert channels with several methods. A plausible strategy is to create confusion by using multiple sources and destinations.

Several "naïve" covert channels can be easily detected by conducting traffic analysis at packet level. Different protocol stack implementations (e.g. different operating systems) normally exhibit well-defined characteristics when generating header fields. These can be used to establish a trusted baseline and identify anomalies that may be a sign of the use of protocol header manipulation for covert communications. An analyst could detect the manipulation of these protocol headers because the values generated by an attacker's tools can be differentiated easily from those generated by a genuine protocol stack. However, if the attacker knows the generation method used by the victim's system, he could encode his data in an identical way, thus creating an undetectable covert channel.

Many of the fields normally chosen by attackers in protocol headers to carry data covertly are designed to contain random values. Actually most of these fields are not completely random, or are only random in a restricted way. When using these fields, covert channels normally try to preserve the randomness to avoid detection, for example by using encryption algorithms before embedding the data. However the range of generated random values is different to the ones generated by the legitimate systems' algorithms, and thus can be detected. In some cases too much randomness can be suspicious.

COVERT CHANNELS – THE FUTURE

It can be safely asserted that the possibilities for establishing covert communications through network covert channels are almost endless. Equally, the research community has invested much effort in identifying covert channels, and is now looking to broaden and diversify this research. Detection targeted

It can be safely asserted that the possibilities for establishing covert communications through network covert channels are almost endless.

specifically on network covert channels is still in its early stages. Most discussions cover the theoretical possibilities of different detection and prevention techniques; many of them are likely to be the central focus of research in the following years.

Some techniques that have been proposed/adapted to detect/prevent covert channels include the use of *Process Query Systems* (PQS), Neural Networks, *Support Vector Machines* (SVM), Quantized Pumps and Active Wardens. Some of these can involve very complex processes, arguably too complex for the risk posed by these threats. Then again, it is no easy task to detect an enemy you can't see, not to mention stopping him getting your organization's sensitive data.*

REFERENCES

[1] Beijtlich, Richard, Powell, G.. The Tao of Network Security Monitoring. Addison Wesley. 2004

[2] Maney, Kevin. "Bin Laden's Messages Could Be Hiding In Plain Sight." USA Today December 19, 2001
http://www.usatoday.com/life/cyber/ccarch/2001/12/19/maney.htm

[3] Petitcolas, Fabien A., Ross J. Anderson and Markus G. Kahn, Information hiding - a Survey. part of IEEE special issue on protection of multimedia content 7/99
http://www.cl.cam.ac.uk/~fapp2/publications/ieee99-infohiding.pdf

[4] Vo2IP Project, Georgia Tech Information Security Center, GA, USA
http://www.voipcc.gtisc.gatech.edu/vo2ip.php

[5] Eu-Jin Goh , Dan Boneh , Benny Pinkas, and Philippe Golle. The Design and Implementation of Protocol-Based Hidden Key Recovery. Stanford University. 2002.

Ron Condon

UK bureau chief
searchsecurity.co.UK

Ron Condon has been writing about developments in the IT industry for more than 30 years. In that time, he has charted the evolution from big mainframes, to minicomputers and PCs in the 1980s, and the rise of the Internet over the last decade or so. In recent years he has specialized in information security. He has edited daily, weekly and monthly publications, and has written for national and regional newspapers, in Europe and the U.S.