# Covert Channels in Internet Protocols: A Survey

D Llamas, C Allison and A Miller
School of Computer Science
University of St Andrews
St Andrews KY16 9SX, Scotland, UK

Tel: +44 (0) 133 4463253
Fax: +44 (0) 133 4463278
{david,ca,alanr}@dcs.st-andrews.ac.uk

*Abstract* – **The creation of covert channels in public computer networks can prove an effective means of information hiding and secret communication. With the widespread adoption of the Internet the TCP/IP suite of protocols have become pervasive, and therefore an attractive target for covert channel exploitation. This paper gives a brief overview of covert channels in communication networks, and presents a brief survey of some recent and relevant papers on the use of covert channels in the common Internet protocols.**

## I. Introduction

A covert channel is a communication channel that allows two cooperating processes to transfer information in a manner that violates the system's security policy [1]. It is thus a way of communicating which is not part of the original design of the system, but can be used to transfer information to a process or user, that, a priori, would not be authorised to access to that information. Covert channels typically only exist in systems with multilevel security [2], which contain and manage information with different sensitivity levels. They allow different users to access to the same information, at the same time, but from different points-of-view, depending on their requirements to know and their access privileges.

The covert channel concept was introduced in 1973 [3]. A classification scheme is proposed in [4]:

- *Scenarios*. In general, when building covert channels, there is a differentiation between *storage* and *timing* covert channels [5]. Storage covert channels are where one process uses direct (or indirect) data writing, whilst another process reads the data. They generally use a finite system resource that is shared between entities with different privileges. Covert timing channels use the modulation of certain resources, such as the CPU timing, in order to exchange information between processes.
- *Noise*. As with any other communication channel, covert channels can be noisy, and vary in their immunity to noise. Ideally, a channel immune to noise is one where the probability of the receiver receiving exactly what the sender has transmitted is unity, and there is no interference in the transmission. Obviously, in real-life, it is very difficult to obtain these perfect channels; hence, it is common to apply error correction codes, which can reduce the bandwidth.
- *Information flows*. With conventional lines of transmission different techniques are applied to increase the bandwidth. A similar method can be achieved in the covert channels. Channels where several information flows are transmitted between sender and receiver are defined as aggregated channels, and depending on how sent variables are initialized, read and reset, aggregations can be classified as serial, parallel, and so on. Channels with a unique information flow are denominated non-aggregated.

The concern for the presence of covert channels is common in high security systems, such as military ones, where typically two observed users know that someone wishes to listen to their conversations. Many of the studies on these attacks, based on covert channels and their prevention, have been done by US government and military bodies, such as the National Security Agency, US Air Force, National Computer Security Centre, and so on.

However, with the dramatic growth of the Internet, there is now a growing concern about the use of covert channels in the TCP/IP protocol suite, which has a number of potential weaknesses that allow an attacker to surreptitiously pass data in otherwise benign packets.

## II. Foundations of covert channels

This survey is based on the analysis of a selection of papers relevant to the use of covert channels in Internet protocols. Although the first three papers are not directly related to the TCP/IP protocols, they have been included due to their general relevance.

### A. *A Guide to Understanding Covert Channel Analysis of Trusted Systems* [4]

This guide has been written to help the vendor and evaluator communities understand the requirements for covert channel analysis as described in the US Department of Defense Trusted Computer System Evaluation Criteria (TCSEC). The guide defines a set of baseline requirements and recommendations for the analysis and evaluation of covert channels. It includes sections focused on the definition and classification of covert channels, identification, bandwidth estimation, covert channel handling, testing and so on.

Covert channels that can only be exploited by security administrators or operators using privileged (i.e. trusted) software are not considered.

### B. *Covert Channels – Here to Stay?* [6]

This paper discusses the difficulties of satisfying high assurance system requirements without sacrificing system capabilities. It also clarifies certain concepts in the theory of covert channels. Traditionally a covert channel's vulnerability was measured by its capacity. It is shown why a capacity analysis alone is not sufficient to evaluate the vulnerability and introduces a new metric referred to as the “small message criterion”.

An overview of covert channel theory is given with examples, and a hypothesis is advanced that covert channels can never be totally eliminated in many "practical" high assurance systems. The paper is organized as follows:

- How reliability and performance requirements can undermine efforts at thwarting covert channels.
- Covert channels in terms of information theory; clarification of certain concepts.
- Capacity analysis alone does not suffice when dealing with covert channels; a new metric referred to as the "small message criterion".
- The trade-offs between covert channel degradation and performance.
- How a middleware buffer reduces the covert channel threat without degrading performance.

### C. *Covert Channel Analysis: A chapter of the Handbook for the Computer Security Certification of Trusted Systems* [7]

This document provides an overview of covert channel analysis, beginning with a definition of covert channels and a discussion of the nature of the issues concerning them that affect the various readers of this handbook. Since covert channels involve (often complex) coding and signalling mechanisms, these are also discussed. The document includes a characterization of covert channels and how the analysis should be performed on system descriptions ranging from abstract models to machine code. Other sections contain issues of system representation and the suitability of various representation paradigms for covert channel analysis. Since covert channels are built from information flows within a trusted computing system, one of the first steps in performing an analysis is the abstraction of potential information flows from a description of the system.

Section 6 focuses on an analysis of covert channels, which includes analyzing threats, channel capacity, possible countermeasures and so on.

## III. Covert channels in internet protocols

### A. *Covert Channels in the TCP/IP Protocol Suite* [8]

Within protocol headers there are many fields that are not used for normal transmission or are "optional", to be set as needed by the sender. This paper illustrates these weaknesses in both theoretical and practical examples. They are about encoding and decoding the following:

- The IP packet identification field.
- The TCP initial sequence number field.
- The TCP acknowledged sequence number field

It includes the C source code for a basic program called covert_tcp, for use on Linux systems, as a proof of the concept.

### B. *Project Loki* [9]

Ping traffic is ubiquitous to almost every TCP/IP based network and subnet. It has a standard packet format recognized by every IP router and is used universally for network management, testing, and measurement. As such, many firewalls and networks consider ping traffic to be benign and will allow it to pass through.

This short but interesting paper explores why that practice can be insecure. Ignoring the obvious threat of the done-to-death denial of service attack, use of ping traffic can open up covert channels through the networks in which it is allowed. This document is intended as a complete description of the covert channels that can exist in networks that allow ICMP_ECHO traffic, to pass. It is a good example of how easily a covert channel can be created based on a very common protocol.

### C. *Ambiguities in TCP/IP* [10]

This paper explores the way that ambiguities in the *implementation* of the TCP/IP suite for various operating systems affect security and covert channels. Although a similar approach has been used for a long time for “OS fingerprinting”, no real attempt has been made yet to identify the security impact of the differences in the TCP/IP semantics.

The paper includes basic research on the TCP “connection open” semantics which is of course very important for security of networked systems. The flaws detected impact on the design of firewalls and packet filters since an improper implementation can easily lead to serious security problems.

### D. *Covert Channel Analysis and Data Hiding in TCP/IP* [11]

This thesis investigates the existence of covert channels in computer networks by analyzing the transport and the Internet layers of the TCP/IP protocol suite. Two approaches for data hiding are identified:

packet header manipulation and packet sorting. Each scenario facilitates the interaction of steganographic principles with the existing network security environment. Specifically, it is shown how associating additional information with IPv4 headers can relax security mechanisms in network nodes such as routers, firewalls, and for services such as authentication, audit, and billing.

Furthermore, the use of packet sorting within the IP Sec framework results in an enhanced network security architecture. While bridging the areas of data hiding, network protocols and network security, both techniques have potential for practical data hiding at the transport and network layers.

Although only a few protocols are considered, this thesis is recommended as many aspects of covert channels are put together within the same scenario. In addition, a novel packet sorting mechanism points to the integration of stego principles with the IPSec architecture.

### E. *Covert Messaging Through TCP Timestamps* [12]

This paper describes a potential covert channel that can be created by manipulating the TCP Timestamp field. By imposing slight delays on the processing of selected TCP packets, the low order bits of their timestamps can be modified. The low bit of the TCP timestamp, when modified in this way, provides a covert channel.

The low bit is effectively random on most connections. Because TCP timestamps are based purely on internal timings of the host, on a slow connection their low bits are randomly distributed. By rewriting the timestamp and varying the timing within the kernel, the value of the low bit can be chosen. As long as values are chosen with a statistically random distribution, they will be indistinguishable from the unaltered values.

Rewriting TCP timestamps presents some additional challenges over and above a standard implementation of the protocol:

- Timestamps must be monotonically increasing
- Timestamps must reflect a reasonable progression of time
- When timestamps are rewritten, it can cause the nonce in the rest of the packet to change

### F. *IP Checksum Covert Channels and Selected Hash Collision* [13]

A fundamental flaw in the design of the Internet checksum, the primary data checksum facility for network data, can allow a malicious user to embed covert channel data in the checksum field itself using a hash collision. What is demonstrated in this paper is the two-way nature of this facility and a covert channel scheme for sending data through the Internet checksum.

This method can be used for any protocol that uses the Internet checksum, including ICMP, UDP, TCP, as well as many others.

It is concluded that the internet checksum is not a secure method for validating data integrity because of the ability of a user to arbitrarily create a selected collision in the hashing mechanism in a trivial period of time.

### G. *Malicious ICMP Tunnelling: Defense Against the Vulnerability* [14]

ICMP is a *required* part of any standards compliant IP node. The paper is organized as follows:

- An introduction to the ICMP tunneling vulnerability
- Standard solutions that can be used to prevent ICMP tunneling
- Results of a modified application using ICMP tunneling
- A proposed solution and its performance impact on routers and on end hosts.

This paper offers some results about one of the most complex areas within the covert channels, which is the real-time manipulation of network packets in order to prevent covert communications through Internet protocols.

### H. *Messaging over IPv6 Destination Options* [15]

This paper is about instant messaging over the IPv6 destination options extension header using the Advanced Sockets API for IPv6 [RFC 2292]. The first 16-bits of each extension header are reserved for the Next Header type that follows, and 8-bits for the header length. The options data can have variable length but must be TLV encoded and aligned to a multiple of 8 octets (IPv6 uses a common format called the Type-Length-Value - TLV - format for variable length fields which are found in the Hop-by-Hop and End-to-End option headers).

The highest-order 2 bits of the Option Type specify the action that must be taken if the option type is not recognized:

- 00 - skip over this option and continue processing the header.
- 01 - discard the packet.

So, all that needs to be done is generate a destination options extension header, TLV encode the message, set the highest-order 2 bits of the option type to 00 and choose an option type value not taken yet.

This is a short but excellent practical paper with an example of code in C about an implementation of covert channels in IPv6.

### I. *Exploitation of data streams authorized by a network access control system for arbitrary data transfers: tunnelling and covert channels over the HTTP protocol* [16]

This paper is about hiding data in the Hypertext Transfer Protocol (HTTP) header and/or body. It is widely known in the security community that the HTTP protocol suffers from a lot of flaws related to the possible use of covert channels. This is understandable as the HTTP protocol was not designed to restrict/protect what researchers wish to present to the community.

This paper also presents an approach to possible scenarios to be taken into account before creating a client/server covert channel tool. They depend on server models, modes and so on. There are interesting sections about using http methods (data containers restrictions, methods with/without message body, the http proxy connect method and so on) and one related to security aspects (authentication, authorization, data stream ciphering, data stream integrity, replay protection and so on).

## IV. Recent works and implementations

### A. *Covert Channel Analysis and Data Hiding in TCP/IP with HTTP Reverse Proxy Servers using Microsoft Windows* [17]

An HTTP forward proxy typically provides Internet access to internal clients that are otherwise restricted by a firewall, and can use caching to reduce network usage. A reverse proxy, by contrast, appears to the client just like an ordinary WWW server, where no special configuration on the client is necessary. The client thus makes ordinary requests for content in the name-space of the reverse proxy. The proxy then decides where to send these requests, and returns the content as if it was the originator.

This paper is about the implementation of covert channels at IP level using HTTP reverse proxy servers, as a transparent element and a middleware component, in order to make detection difficult.

An implementation of covert channels under Microsoft Windows platforms is described In all the Windows operating systems, the TCP/IP protocol implementation is proprietary, and its source code is not accessible which means that the manipulation of the packets is not possible from levels above the TCP/IP driver layer. This makes the use of these techniques in a Windows platform more complex.

A list of network traffic filtering technologies for Windows (user and kernel modes) is also provided. Those technologies affect the possible covert channels that can be created in Internet protocols under this platform.

### B. *IP Covert Timing Channels: Design and Detection* [18]

In this paper, an implementation of a covert network timing channel is described, the subtle issues that arose in its design are discussed, and performance data for the channel is presented.

The implementation is used as the basis for some experiments in its detection. It is shown that the regularity of a timing channel can be used to differentiate it from other traffic. Two methods of doing so and measures of their efficiency are described. Mechanisms that attackers might use to disrupt the regularity of the timing channel are investigated, and methods of detection that are effective against them are demonstrated (simple timing channel, varying the timing interval, injecting noise).

This paper is an excellent example of covert timing channels, which are usually more complex types than the ones based on storage.

### C. *Embedding Covert Channels into TCP/IP* [19]

It is commonly believed that steganography within TCP/IP is easily achieved by embedding data in header fields seemingly filled with “random” data, such as the IP identifier, TCP initial sequence number or the least significant bit of the TCP timestamp.

The authors of this paper show that this is not the case; these fields naturally exhibit sufficient structure and non-uniformity to be efficiently and reliably differentiated from unmodified cipher text. Previous work on TCP/IP steganography does not take this into account and, by examining TCP/IP specifications and open source implementations, the authors have developed tests to detect the use of these embedding techniques.

A detailed description of the TCP Initial Sequence Number (ISN) and IP identifier generation schemes in Linux and OpenBSD is presented, and a number of previously proposed schemes for TCP/IP-based steganography are described. It is shown that a passive warden can detect the use of these schemes because the modified headers that they produce can be distinguished from those generated by a genuine TCP/IP stack.

Finally, two schemes are outlined for encoding data with ISNs generated by OpenBSD and Linux. Both schemes generate ISNs that are indistinguishable from those generated by a genuine TCP stack, except by those with knowledge of a shared secret key.

## V. Conclusions

This paper has provided an overview of relevant papers about the theory and practice of covert channels in common Internet protocols. It has shown how features of these protocols can be used to store hidden information and how covert communications can be produced by considering operational aspects, such as timing.

Another point that arises from this survey is how theory is a more consolidated aspect than practice. This can be due to the fact that there are multiple implementations of the TCP/IP suite of protocols, and it is not always clear how a covert channel will work within a particular scenario and if those results will be able to be extrapolated.

Finally, this survey contributes to the reflection that the combination of steganography techniques and covert channels constitute a more complex scenario to be taken into account in the future, as Internet protocols continue to evolve in response to their increasingly widespread usage.

## ACKNOWLEDGMENT

This work has been performed in the framework of the PhD research programme of David Llamas at the University of St Andrews, which is supported by the Defence Science and Technology Laboratory - DSTL - a part of the Ministry of Defence of the United Kingdom.

## REFERENCES

[1] S. Berg, "Glossary of Computer Security Terms." National Computer Security Center, USA, http://zedz.net/rainbow/NCSC-TG-004.ps 1998.

[2] N. E. Proctor and P. G. Neumann, "Architectural Implications of Covert Channels." Computer Science Lab, SRI International, USA, http://www.csl.sri.com/users/neumann/ncs92.html 1992.

[3] B. W. Lampson, "A Note on the Confinement Problem." Xerox Palo Alto Research Center, USA, http://www.cis.upenn.edu/~KeyKOS/Confinement.html 1973.

[4] P. R. Gallagher, "A Guide to Understanding Covert Channel Analysis of Trusted Systems." National Computer Security Center, USA, http://fas.org/irp/nsa/rainbow/tg030.htm 1993.

[5] S. B. Lipner, "A Comment on the Confinement Problem." MITRE Corporation, USA, Proceedings of the fifth ACM symposium on Operating systems principles, ACM Press 1975.

[6] I. S. Moskowitz and M. H. Kang, "Covert Channels - Here to Stay?." Naval Research Laboratory, Department of the Navy, USA, http://chacs.nrl.navy.mil/publications/CHACS/1994moskowitz-compass.ps 1994.

[7] J. McHugh, "Covert Channel Analysis: A chapter of the Handbook for the Computer Security Certification of Trusted Systems." Department of Computer Science, Portland State University, USA, http://chacs.nrl.navy.mil/publications/handbook 1995.

[8] C. H. Rowland, "Covert channels in the TCP/IP protocol suite." First Monday, USA, http://www.firstmonday.dk/issues/issue2_5/rowland/ 1996.

[9] daemon9, AKA, and route, "Project Loki," in *Volume Seven*, *Issue Forty-Nine*. Phrack Magazine, USA, http://www.phrack.org/show.php?p=49&a=6 1996.

[10] P. Starzetz, "Ambiguities in TCP/IP." Securityfocus, University of Stuttgart, German, http://gray-world.net/fr/papers/ambiguitiesintcpip.txt 2002.

[11] K. Ahsan, "Covert Channel Analysis and Data Hiding in TCP/IP." University of Toronto, Canada, http://gray-world.net/papers/ahsan02.pdf 2002.

[12] J. Giffin, R. Greenstadt, P. Litwack, and R. Tibbetts, "Covert Messaging Through TCP Timestamps." Massachusetts Institute of Technology - MIT, USA, http://web.mit.edu/greenie/Public/CovertMessagingi nTCP.ps 2002.

[13] C. Abad, "IP Checksum Covert Channels and Selected Hash Collision." University of California, USA, http://downloads.securityfocus.com/library/ipccc.pdf 2001.

[14] A. Singh, O. Nordstrom, C. Lu, and A. Santos, "Malicious ICMP Tunnelling: Defense Against the Vulnerability." Center for Experimental Research in Computer Systems, Georgia Institute of Technology, USA, http://www.cc.gatech.edu/~abhi/icmp-paper.ps 2003.

[15] T. Graf, "Messaging over IPv6 Destination Options." The Swiss Unix User Group, Switzerland, http://gray-world.net/papers/messip6.txt 2003.

[16] A. Dyatlov and S. Castro, "Exploitation of data streams authorized by a network access control system for arbitrary data transfers: tunnelling and covert channels over the HTTP protocol." Gray-world, USA, http://gray-world.net/projects/papers/html/covert_paper.html 2003.

[17] D. Llamas and W. Buchanan, "Covert Channel Analysis and Data Hiding in TCP/IP with Reverse Proxy Servers using Microsoft Windows." 3rd European Conference On Information Warfare and Security, University of London, UK, http://www.davidllamas.org/academic%20resources/0406-ECIW/ECIW04-Paper.pdf 2004.

[18] S. Cabuk, C. Brodley, and C. Shields, "IP Covert Timing Channels: Design and Detection." Purdue University, Tufts University, Georgetown University, USA, http://www.cs.jhu.edu/~fabian/courses/CS600.624/covert.pdf 2004.

[19] S. J. Murdoch and S. Lewis, "Embedding Covert Channels into TCP/IP." Computer Laboratory, University of Cambridge, UK, http://www.cl.cam.ac.uk/~sjm217/papers/ih05coverttcp.pdf 2005.