TCP Maintenance and Minor
Extensions (tcpm)
Internet-Draft
Intended status: BCP
Expires: February 20, 2010

F. Gont
UK CPNI

August 19, 2009

Security Assessment of the Transmission Control Protocol (TCP)
draft-ietf-tcpm-tcp-security-00.txt

Status of this Memo

Copyright Notice

Abstract

This document contains a security assessment of the specifications of the Transmission Control Protocol (TCP), and of a number of

mechanisms and policies in use by popular TCP implementations. Additionally, it contains best current practices for hardening a TCP implementation.

Table of Contents

1. Preface

1.1. Introduction

The TCP/IP protocol suite was conceived in an environment that was quite different from the hostile environment they currently operate in. However, the effectiveness of the protocols led to their early adoption in production environments, to the point that, to some extent, the current world's economy depends on them.

While many textbooks and articles have created the myth that the Internet protocols were designed for warfare environments, the top level goal for the DARPA Internet Program was the sharing of large service machines on the ARPANET [Clark, 1988]. As a result, many protocol specifications focus only on the operational aspects of the protocols they specify, and overlook their security implications.

While the Internet technology evolved since it early inception, the Internet's building blocks are basically the same core protocols adopted by the ARPANET more than two decades ago. During the last twenty years, many vulnerabilities have been identified in the TCP/IP stacks of a number of systems. Some of them were based on flaws in

some protocol implementations, affecting only a reduced number of systems, while others were based in flaws in the protocols themselves, affecting virtually every existing implementation [Bellovin, 1989]. Even in the last couple of years, researchers were still working on security problems in the core protocols [NISCC, 2004] [NISCC, 2005].

The discovery of vulnerabilities in the TCP/IP protocol suite usually led to reports being published by a number of CSIRTs (Computer Security Incident Response Teams) and vendors, which helped to raise awareness about the threats and the best mitigations known at the time the reports were published. Unfortunately, this also led to the documentation of the discovered protocol vulnerabilities being spread among a large number of documents, which are sometimes difficult to identify.

For some reason, much of the effort of the security community on the Internet protocols did not result in official documents (RFCs) being issued by the IETF (Internet Engineering Task Force). This basically led to a situation in which "known" security problems have not always been addressed by all vendors. In addition, in many cases vendors have implemented quick "fixes" to the identified vulnerabilities without a careful analysis of their effectiveness and their impact on interoperability [Silbersack, 2005].

Producing a secure TCP/IP implementation nowadays is a very difficult

task, in part because of the lack of a single document that serves as a security roadmap for the protocols. Implementers are faced with the hard task of identifying relevant documentation and differentiating between that which provides correct advice, and that which provides misleading advice based on inaccurate or wrong assumptions.

This document is the result of a security assessment of the IETF specifications of the Transmission Control Protocol (TCP), from a security point of view. Possible threats are identified and, where possible, countermeasures are proposed. Additionally, many implementation flaws that have led to security vulnerabilities have been referenced in the hope that future implementations will not incur the same problems.

This document is heavily based on the "Security Assessment of the Transmission Control Protocol (TCP)" released by the UK Centre for the Protection of National Infrastructure (CPNI), available at: http://www.cpni.gov.uk/Products/technicalnotes/Feb-09-security-assessment-TCP.aspx .

1.2. Scope of this document

While there are a number of protocols that may affect the way TCP operates, this document focuses only on the specifications of the Transmission Control Protocol (TCP) itself.

The following IETF RFCs were selected for assessment as part of this work:

- RFC 793, "Transmission Control Protocol. DARPA Internet Program. Protocol Specification" (91 pages)
- RFC 1122, "Requirements for Internet Hosts -- Communication Layers" (116 pages)
- RFC 1191, "Path MTU Discovery" (19 pages)
- RFC 1323, "TCP Extensions for High Performance" (37 pages)
- RFC 1948, "Defending Against Sequence Number Attacks" (6 pages)
- RFC 1981, "Path MTU Discovery for IP version 6" (15 pages)
- RFC 2018, "TCP Selective Acknowledgment Options" (12 pages)
- RFC 2385, "Protection of BGP Sessions via the TCP MD5 Signature Option" (6 pages)

- RFC 2581, "TCP Congestion Control" (14 pages)
- RFC 2675, "IPv6 Jumbograms" (9 pages)
- RFC 2883, "An Extension to the Selective Acknowledgement (SACK) Option for TCP" (17 pages)
- RFC 2884, "Performance Evaluation of Explicit Congestion

Notification (ECN) in IP Networks" (18 pages)

- RFC 2988, "Computing TCP's Retransmission Timer" (8 pages)
- RFC 3168, "The Addition of Explicit Congestion Notification (ECN) to IP" (63 pages)
- RFC 3465, "TCP Congestion Control with Appropriate Byte Counting (ABC)" (10 pages)
- RFC 3517, "A Conservative Selective Acknowledgment (SACK)-based Loss Recovery Algorithm for TCP" (13 pages)
- RFC 3540, "Robust Explicit Congestion Notification (ECN) Signaling with Nonces" (13 pages)
- RFC 3782, "The NewReno Modification to TCP's Fast Recovery Algorithm" (19 pages)

1.3. Organization of this document

This document is basically organized in two parts. The first part contains a discussion of each of the TCP header fields, identifies their security implications, and discusses the possible countermeasures. The second part contains an analysis of the security implications of the mechanisms and policies implemented by TCP, and of a number of implementation strategies in use by a number of popular TCP implementations.

2. The Transmission Control Protocol

The Transmission Control Protocol (TCP) is a connection-oriented transport protocol that provides a reliable byte-stream data transfer service.

Very few assumptions are made about the reliability of underlying data transfer services below the TCP layer. Basically, TCP assumes it can obtain a simple, potentially unreliable datagram service from the lower level protocols. Figure 1 illustrates where TCP fits in

the DARPA reference model.

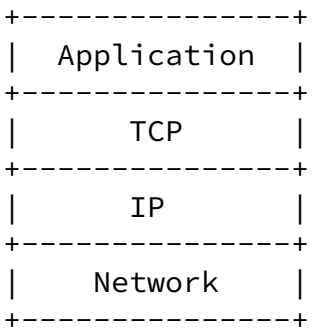


Figure 1: TCP in the DARPA reference model

TCP provides facilities in the following areas:

- Basic Data Transfer
- Reliability
- Flow Control
- Multiplexing
- Connections
- Precedence and Security
- Congestion Control

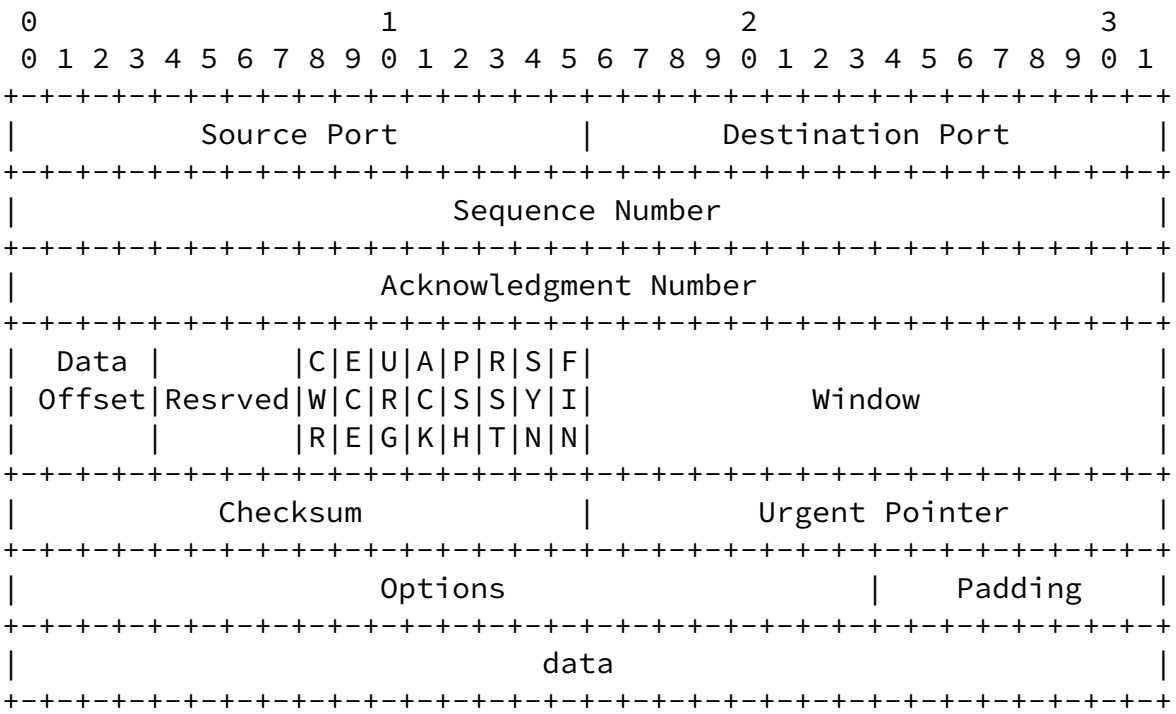
The core TCP specification, RFC 793 [Postel, 1981c], dates back to 1981 and standardizes the basic mechanisms and policies of TCP. RFC 1122 [Braden, 1989] provides clarifications and errata for the original specification. RFC 2581 [Allman et al, 1999] specifies TCP congestion control and avoidance mechanisms, not present in the original specification. Other documents specify extensions and improvements for TCP.

The large amount of documents that specify extensions, improvements, or modifications to existing TCP mechanisms has led the IETF to publish a roadmap for TCP, RFC 4614 [Duke et al, 2006], that clarifies the relevance of each of those documents.

3. TCP header fields

RFC 793 [Postel, 1981c] defines the syntax of a TCP segment, along with the semantics of each of the header fields. Figure 2

illustrates the syntax of a TCP segment.



Note that one tick mark represents one bit position

Figure 2: Transmission Control Protocol header format

The minimum TCP header size is 20 bytes, and corresponds to a TCP segment with no options and no data. However, a TCP module might be handed an (illegitimate) "TCP segment" of less than 20 bytes. Therefore, before doing any processing of the TCP header fields, the following check should be performed by TCP on the segments handed by the internet layer:

Segment.Size >= 20

If a segment does not pass this check, it should be dropped.

The following subsections contain further sanity checks that should be performed on TCP segments.

3.1. Source Port

This field contains a 16-bit number that identifies the TCP end-point that originated this TCP segment. Being a 16-bit field, it can contain any value in the range 0-65535.

The Internet Assigned Numbers Authority (IANA) has traditionally

reserved the following use of the 16-bit port range of TCP [IANA, 2008]:

- The Well Known Ports, 0 through 1023
- The Registered Ports, 1024 through 49151
- The Dynamic and/or Private Ports, 49152 through 65535

The range of assigned ports managed by the IANA is 0-1023, with the remainder being registered by IANA but not assigned [IANA, 2008]. It is also worth noting that, while some systems restrict use of the port numbers in the range 0-1024 to privileged users, no trust should be granted based on the port numbers used for a TCP connection.

Servers usually bind specific ports on which specific services are usually provided, while clients usually make use of the so-called "ephemeral ports" for the source port of their outgoing connections with the only requirement that the resulting four-tuple must be unique (not currently in use by any other transport protocol instance).

While the only requirement for a selected ephemeral port is that the resulting four-tuple (connection-id) is unique, in practice it may be necessary to not allow the allocation of port numbers that are in use by a TCP that is in the LISTEN or CLOSED states for use as ephemeral ports, as this might allow an attacker to "steal" incoming connections from a local server application. Section 10.2 of this document provides a detailed discussion of this issue.

It should also be noted that some clients, such as DNS resolvers, are known to use port numbers from the "Well Known Ports" range. Therefore, middle-boxes such as packet filters should not assume that clients use port number from only the Dynamic or Registered port ranges.

While port 0 is a legitimate port number, it has a special meaning in the UNIX Sockets API. For example, when a TCP port number of 0 is passed as an argument to the bind() function, rather than binding

port 0, an ephemeral port is selected for the corresponding TCP end-point. As a result, the TCP port number 0 is never actually used in TCP segments.

Different implementations have been found to respond differently to TCP segments that have a port number of 0 as the Source Port and/or the Destination Port. As a result, TCP segments with a port number of 0 are usually employed for remote OS detection via TCP/IP stack fingerprinting [Jones, 2003].

Since in practice TCP port 0 is not used by any legitimate application and is only used for fingerprinting purposes, a number of host implementations already reject TCP segments that use 0 as the Source Port and/or the Destination Port. Also, a number firewalls filter (by default) any TCP segments that contain a port number of zero for the Source Port and/or the Destination Port.

We therefore recommend that TCP implementations respond to incoming TCP segments that have a Source Port of 0 with an RST (provided these incoming segments do not have the RST bit set).

Responding with an RST segment to incoming segments that have the RST bit would open the door to RST-war attacks.

As discussed in Section 3.2, we also recommend TCP implementations to respond with an RST to incoming packets that have a Destination Port of 0 (provided these incoming segments do not have the RST bit set).

3.1.1. Problems that may arise as a result of collisions of connection-id's

A number of implementations will not allow the creation of a new connection if there exists a previous incarnation of the same connection in any state other than the fictional state CLOSED. This can be problematic in scenarios in which a client establishes connections with a specific service at a particular server at a high rate: even if the connections are also closed at a high rate, one of the systems (the one performing the active close) will keep each of the closed connections in the TIME-WAIT state for 2*MSL.

MSL (Maximum Segment Lifetime) is the maximum amount of time that a TCP segment can exist in an internet. It is defined to be 2 minutes

by RFC 793 [Postel, 1981c].

If the connection rate is high enough, at some point all the ephemeral ports at the client will be in use by some connection in the TIME-WAIT state, thus preventing the establishment of new connections. In order to overcome this problem, a number of TCP implementations include some heuristics to allow the creation of a new incarnation of a connection that is in the TIME-WAIT state. In such implementations a new incarnation of a previous connection is allowed if:

- The incoming SYN segment contains a timestamp option, and the timestamp is greater than the last timestamp seen in the previous incarnation of the connection (for that direction of the data transfer), or,

- The incoming SYN segment does not contain a timestamp option, but its Initial Sequence Number (ISN) is greater than the last sequence number seen in the previous incarnation of the connection (for that direction of the data transfer)

Unfortunately, these heuristics are optional, and thus cannot be relied upon. Additionally, as indicated by [Silbersack, 2005], if the Timestamp or the ISN are trivially randomized, these heuristics might fail.

Section 3.3.1 and Section 4.7.1 of this document recommend algorithms for the generation of TCP Initial Sequence Numbers and TCP timestamps, respectively, that provide randomization, while still allowing the aforementioned heuristics to work.

Therefore, the only strategy that can be relied upon to avoid this interoperability problem is to minimize the rate of collisions of connection-id's. A good algorithm to minimize rate of collisions of connection-id's would consider the time a given four-tuple {Source Address, Source Port, Destination Address, Destination Port} was last used, and would try avoid reusing it for 2*MSL. However, an efficient implementation approach for this algorithm has not yet been devised. A simple approach to minimize the rate collisions of connection-id's in most scenarios is to maximize the port reuse cycle, such that a port number is not reused before all the other

port numbers in the ephemeral port range have been used for outgoing connections. This is the traditional ephemeral port selection algorithm in 4.4BSD implementations.

However, if a single global variable is used to keep track of the last ephemeral port selected, ephemeral port numbers become trivially predictable.

Section 3.1.2 of this document analyzes a number of approaches for obfuscating the TCP ephemeral ports, such that the chances of an attacker of guessing the ephemeral ports used for future connections are reduced, while still reducing the probability of collisions of connection-id's. Finally, Section 3.1.3 makes recommendations about the port range that should be used for the ephemeral ports.

16.3. Broadcast and multicast IP addresses

17. Security Considerations

18. Acknowledgements

This document is based on the document "Security Assessment of the Transmission Control Protocol (TCP)" [CPNI, 2009] written by Fernando Gont on behalf of CPNI (Centre for the Protection of National Infrastructure).

The author would like to thank (in alphabetical order) Randall Atkinson, Guillermo Gont, Alfred Hoenes, Jamshid Mahdavi, Stanislav Shalunov, Michael Welzl, Dan Wing, Andrew Yourtchenko, Michael Zalewski, and Christos Zoulas, for providing valuable feedback on earlier versions of the UK CPNI document.

Additionally, the author would like to thank (in alphabetical order) Mark Allman, David Black, Ethan Blanton, David Borman, James Chacon, John Heffner, Jerrold Leichter, Jamshid Mahdavi, Keith Scott, Bill Squier, and David White, who generously answered a number of questions that araised while the aforementioned document was being written.

Finally, the author would like to thank CPNI (formely NISCC) for their continued support.

19. References

Abley, J., Savola, P., Neville-Neil, G. 2007. Deprecation of Type 0 Routing Headers in IPv6. RFC 5095.

Allman, M. 2003. TCP Congestion Control with Appropriate Byte Counting (ABC). RFC 3465.

Allman, M. 2008. Comments On Selecting Ephemeral Ports. Available

at: http://www.icir.org/mallman/share/ports-dec08.pdf

Allman, M., Paxson, V., Stevens, W. 1999. TCP Congestion Control. RFC 2581.

Allman, M., Balakrishnan, H., Floyd, S. 2001. Enhancing TCP's Loss Recovery Using Limited Transmit. RFC 3042.

Allman, M., Floyd, S., and C. Partridge. 2002. Increasing TCP's Initial Window. RFC 3390.

Baker, F. 1995. Requirements for IP Version 4 Routers. RFC 1812.

Baker, F., Savola, P. 2004. Ingress Filtering for Multihomed Networks. RFC 3704.

Barisani, A. 2006. FTester - Firewall and IDS testing tool. Available at: http://dev.inversepath.com/trac/ftester

Beck, R. 2001. Passive-Aggressive Resistance: OS Fingerprint Evasion. Linux Journal.

Bellovin, S. M. 1989. Security Problems in the TCP/IP Protocol Suite. Computer Communication Review, Vol. 19, No. 2, pp. 32-48.

Bellovin, S. M. 1996. Defending Against Sequence Number Attacks. RFC 1948.

Bellovin, S. M. 2006. Towards a TCP Security Option. IETF Internet-Draft (draft-bellovin-tcpsec-00.txt), work in progress.

Bernstein, D. J. 1996. SYN cookies. Available at: http://cr.yp.to/syncookies.html

Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and Weiss, W., 1998. An Architecture for Differentiated Services. RFC 2475.

Blanton, E., Allman, M., Fall, K., Wang, L. 2003. A Conservative Selective Acknowledgment (SACK)-based Loss Recovery Algorithm for TCP. RFC 3517.

Borman, D. 1997. Post to the tcp-impl mailing-list. Message-Id:

<199706061526.KAA01535@frantic.BSDI.COM>. Available at: http://www.kohala.com/start/borman.97jun06.txt

Borman, D., Deering, S., Hinden, R. 1999. IPv6 Jumbograms. RFC 2675.

Braden, R. 1989. Requirements for Internet Hosts -- Communication Layers. RFC 1122.

Braden, R. 1992. Extending TCP for Transactions -- Concepts. RFC 1379.

Braden, R. 1994. T/TCP -- TCP Extensions for Transactions Functional Specification. RFC 1644.

CCSDS. 2006. Consultative Committee for Space Data Systems (CCSDS) Recommendation Communications Protocol Specification (SCPS) -- Transport Protocol (SCPS-TP). Blue Book. Issue 2. Available at: http://public.ccsds.org/publications/archive/714x0b2.pdf

CERT. 1996. CERT Advisory CA-1996-21: TCP SYN Flooding and IP Spoofing Attacks. Available at:

http://www.cert.org/advisories/CA-1996-21.html

CERT. 1997. CERT Advisory CA-1997-28 IP Denial-of-Service Attacks. Available at: http://www.cert.org/advisories/CA-1997-28.html

CERT. 2000. CERT Advisory CA-2000-21: Denial-of-Service Vulnerabilities in TCP/IP Stacks. Available at: http://www.cert.org/advisories/CA-2000-21.html

CERT. 2001. CERT Advisory CA-2001-09: Statistical Weaknesses in TCP/IP Initial Sequence Numbers. Available at: http://www.cert.org/advisories/CA-2001-09.html

CERT. 2003. CERT Advisory CA-2003-13 Multiple Vulnerabilities in Snort Preprocessors. Available at: http://www.cert.org/advisories/CA-2003-13.html

Cisco. 2008a. Cisco Security Appliance Command Reference, Version 7.0. Available at: http://www.cisco.com/en/US/docs/security/asa/

asa70/command/reference/tz.html#wp1288756

Cisco. 2008b. Cisco Security Appliance System Log Messages, Version 8.0. Available at: http://www.cisco.com/en/US/docs/security/asa/asa80/system/message/logmsgs.html#wp4773952

Clark, D.D. 1982. Fault isolation and recovery. RFC 816.

Clark, D.D. 1988. The Design Philosophy of the DARPA Internet Protocols, Computer Communication Review, Vol. 18, No.4, pp. 106-114.

Connolly, T., Amer, P., Conrad, P. 1994. An Extension to TCP : Partial Order Service. RFC 1693.

Conta, A., Deering, S., Gupta, M. 2006. Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification. RFC 4443.

CORE. 2003. Core Secure Technologies Advisory CORE-2003-0307: Snort TCP Stream Reassembly Integer Overflow Vulnerability. Available at: http://www.coresecurity.com/common/showdoc.php?idx=313&idxseccion=10

CPNI, 2008. Security Assessment of the Internet Protocol. Available at: http://www.cpni.gov.uk/Docs/InternetProtocol.pdf

CPNI, 2009. Security Assessment of the Transmission Control Protocol (TCP). Available at: http://www.cpni.gov.uk/Docs/tn-03-09-security-assessment-TCP.pdf

daemon9, route, and infinity. 1996. IP-spoofing Demystified (Trust-Relationship Exploitation), Phrack Magazine, Volume Seven, Issue Forty-Eight, File 14 of 18. Available at: http://www.phrack.org/archives/48/P48-14

Deering, S., Hinden, R. 1998. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460.

Dharmapurikar, S., Paxson, V. 2005. Robust TCP Stream Reassembly In the Presence of Adversaries. Proceedings of the USENIX Security Symposium 2005.

Duke, M., Braden, R., Eddy, W., Blanton, E. 2006. A Roadmap for Transmission Control Protocol (TCP) Specification Documents. RFC 4614.

Ed3f. 2002. Firewall spotting and networks analisys with a broken CRC. Phrack Magazine, Volume 0x0b, Issue 0x3c, Phile #0x0c of 0x10. Available at: http://www.phrack.org/phrack/60/p60-0x0c.txt

Eddy, W. 2007. TCP SYN Flooding Attacks and Common Mitigations. RFC 4987.

Fenner, B. 2006. Experimental Values in IPv4, IPv6, ICMPv4, ICMPv6, UDP, and TCP Headers. RFC 4727.

Ferguson, P., and Senie, D. 2000. Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing. RFC 2827.

Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and Berners-Lee, T. 1999. Hypertext Transfer Protocol -- HTTP/1.1. RFC 2616.

Floyd, S., Mahdavi, J., Mathis, M., Podolsky, M. 2000. An Extension to the Selective Acknowledgement (SACK) Option for TCP. RFC 2883.

Floyd, S., Henderson, T., Gurtov, A. 2004. The NewReno Modification to TCP's Fast Recovery Algorithm. RFC 3782.

Floyd, S., Allman, M., Jain, A., Sarolahti, P. 2007. Quick-Start for TCP and IP. RFC 4782.

Fyodor. 1998. Remote OS Detection via TCP/IP Stack Fingerprinting. Phrack Magazine, Volume 8, Issue, 54.

Fyodor. 2006a. Remote OS Detection via TCP/IP Fingerprinting (2nd Generation). Available at: http://insecure.org/nmap/osdetect/.

Fyodor. 2006b. Nmap - Free Security Scanner For Network Exploration and Audit. Available at: http://www.insecure.org/nmap.

Fyodor. 2008. Nmap Reference Guide: Port Scanning Techniques. Available at: http://nmap.org/book/man-port-scanning-techniques.html

GIAC. 2000. Egress Filtering v 0.2. Available at: http://www.sans.org/y2k/egress.htm

Giffin, J., Greenstadt, R., Litwack, P., Tibbetts, R. 2002. Covert Messaging through TCP Timestamps. PET2002 (Workshop on Privacy Enhancing Technologies), San Francisco, CA, USA, April 2002. Available at: http://web.mit.edu/greenie/Public/CovertMessaginginTCP.ps

Gill, V., Heasley, J., Meyer, D., Savola, P, Pignataro, C. 2007. The Generalized TTL Security Mechanism (GTSM). RFC 5082.

Gont, F. 2006. Advanced ICMP packet filtering. Available at: http://www.gont.com.ar/papers/icmp-filtering.html

Gont, F. 2008a. ICMP attacks against TCP. IETF Internet-Draft (draft-ietf-tcpm-icmp-attacks-04.txt), work in progress.

Gont, F.. 2008b. TCP's Reaction to Soft Errors. IETF Internet-Draft (draft-ietf-tcpm-tcp-soft-errors-09.txt), work in progress.

Gont, F. 2009. On the generation of TCP timestamps. IETF Internet-Draft (draft-gont-tcpm-tcp-timestamps-01.txt), work in progress.

Gont, F., Srisuresh, P. 2008. Security Implications of Network Address Translators (NATs). IETF Internet-Draft (draft-gont-behave-nat-security-01.txt), work in progress.

Gont, F., Yourtchenko, A. 2009. On the implementation of TCP urgent data. IETF Internet-Draft (draft-gont-tcpm-urgent-data-01.txt), work in progress.

Heffernan, A. 1998. Protection of BGP Sessions via the TCP MD5 Signature Option. RFC 2385.

Heffner, J. 2002. High Bandwidth TCP Queuing. Senior Thesis.

Hoenes, A. 2007. TCP options - tcp-parameters IANA registry. Post to the tcpm wg mailing-list. Available at: http://www.ietf.org/mail-archive/web/tcpm/current/msg03199.html

IANA. 2007. Transmission Control Protocol (TCP) Option Numbers.

Avialable at: http://www.iana.org/assignments/tcp-parameters/

IANA. 2008. Port Numbers. Available at: http://www.iana.org/assignments/port-numbers

Jacobson, V. 1988. Congestion Avoidance and Control. Computer Communication Review, vol. 18, no. 4, pp. 314-329. Available at: ftp://ftp.ee.lbl.gov/papers/congavoid.ps.Z

Jacobson, V., Braden, R. 1988. TCP Extensions for Long-Delay Paths. RFC 1072.

Jacobson, V., Braden, R., Borman, D. 1992. TCP Extensions for High Performance. RFC 1323.

Jones, S. 2003. Port 0 OS Fingerprinting. Available at: http://www.gont.com.ar/docs/port-0-os-fingerprinting.txt

Kent, S. and Seo, K. 2005. Security Architecture for the Internet Protocol. RFC 4301.

Klensin, J. 2008. Simple Mail Transfer Protocol. RFC 5321.

Ko, Y., Ko, S., and Ko, M. 2001. NIDS Evasion Method named SeolMa. Phrack Magazine, Volume 0x0b, Issue 0x39, phile #0x03 of 0x12. Available at: http://www.phrack.org/issues.html?issue=57&id=3#article

Lahey, K. 2000. TCP Problems with Path MTU Discovery. RFC 2923.

Larsen, M., Gont, F. 2008. Port Randomization. IETF Internet-Draft (draft-ietf-tsvwg-port-randomization-02), work in progress.

Lemon, 2002. Resisting SYN flood DoS attacks with a SYN cache. Proceedings of the BSDCon 2002 Conference, pp 89-98.

Maimon, U. 1996. Port Scanning without the SYN flag. Phrack Magazine, Volume Seven, Issue Fourty-Nine, phile #0x0f of 0x10. Available at: http://www.phrack.org/issues.html?issue=49&id=15#article

Mathis, M., Mahdavi, J., Floyd, S. Romanow, A. 1996. TCP Selective Acknowledgment Options. RFC 2018.

Mathis, M., and Heffner, J. 2007. Packetization Layer Path MTU Discovery. RFC 4821.

McCann, J., Deering, S., Mogul, J. 1996. Path MTU Discovery for IP version 6. RFC 1981.

McKusick, M., Bostic, K., Karels, M., and J. Quarterman. 1996. The Design and Implementation of the 4.4BSD Operating System. Addison-Wesley.

Meltman. 1997. new TCP/IP bug in win95. Post to the bugtraq mailing-list. Available at: http://insecure.org/sploits/land.ip.DOS.html

Miller, T. 2006. Passive OS Fingerprinting: Details and Techniques. Available at: http://www.ouah.org/incosfingerp.htm .

Mogul, J., and Deering, S. 1990. Path MTU Discovery. RFC 1191.

Morris, R. 1985. A Weakness in the 4.2BSD Unix TCP/IP Software. Technical Report CSTR-117, AT&T Bell Laboratories. Available at: http://pdos.csail.mit.edu/~rtm/papers/117.pdf .

Myst. 1997. Windows 95/NT DoS. Post to the bugtraq mailing-list. Available at: http://seclists.org/bugtraq/1997/May/0039.html

Nichols, K., Blake, S., Baker, F., and Black, D. 1998. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. RFC 2474.

NISCC. 2004. NISCC Vulnerability Advisory 236929: Vulnerability Issues in TCP. Available at: http://www.uniras.gov.uk/niscc/docs/re-20040420-00391.pdf

NISCC. 2005. NISCC Vulnerability Advisory 532967/NISCC/ICMP: Vulnerability Issues in ICMP packets with TCP payloads. Available at: http://www.niscc.gov.uk/niscc/docs/re-20050412-00303.pdf

NISCC. 2006. NISCC Technical Note 01/2006: Egress and Ingress Filtering. Available at: http://www.niscc.gov.uk/niscc/docs/re-20060420-00294.pdf?lang=en

Ostermann, S. 2008. tcptrace tool. Tool and documentation available at: http://www.tcptrace.org.

Paxson, V., Allman, M. 2000. Computing TCP's Retransmission Timer. RFC 2988.

PCNWG. 2009. Congestion and Pre-Congestion Notification (pcn)

charter. Available at:
http://www.ietf.org/html.charters/pcn-charter.html

PMTUDWG. 2007. Path MTU Discovery (pmtud) charter. Available at:
http://www.ietf.org/html.charters/OLD/pmtud-charter.html

Postel, J. 1981a. Internet Protocol. DARPA Internet Program. Protocol Specification. RFC 791.

Postel, J. 1981b. Internet Control Message Protocol. RFC 792.

Postel, J. 1981c. Transmission Control Protocol. DARPA Internet Program. Protocol Specification. RFC 793.

Postel, J. 1987. TCP AND IP BAKE OFF. RFC 1025.

Ptacek, T. H., and Newsham, T. N. 1998. Insertion, Evasion and Denial of Service: Eluding Network Intrusion Detection. Secure Networks, Inc. Available at:
http://www.aciri.org/vern/Ptacek-Newsham-Evasion-98.ps

Ramaiah, A., Stewart, R., and Dalal, M. 2008. Improving TCP's Robustness to Blind In-Window Attacks. IETF Internet-Draft (draft-ietf-tcpm-tcpsecure-10.txt), work in progress.

Ramakrishnan, K., Floyd, S., and Black, D. 2001. The Addition of Explicit Congestion Notification (ECN) to IP. RFC 3168.

Rekhter, Y., Li, T., Hares, S. 2006. A Border Gateway Protocol 4 (BGP-4). RFC 4271.

Rivest, R. 1992. The MD5 Message-Digest Algorithm. RFC 1321.

Rowland, C. 1997. Covert Channels in the TCP/IP Protocol Suite. First Monday Journal, Volume 2, Number 5. Available at:
http://www.firstmonday.org/issues/issue2_5/rowland/

Savage, S., Cardwell, N., Wetherall, D., Anderson, T. 1999. TCP Congestion Control with a Misbehaving Receiver. ACM Computer Communication Review, 29(5), October 1999.

Semke, J., Mahdavi, J., Mathis, M. 1998. Automatic TCP Buffer Tuning. ACM Computer Communication Review, Vol. 28, No. 4.

Shalunov, S. 2000. Netkill. Available at: http://www.internet2.edu/~shalunov/netkill/netkill.html

Shimomura, T. 1995. Technical details of the attack described by Markoff in NYT. Message posted in USENET's comp.security.misc newsgroup, Message-ID: <3g5gkl$5j1@ariel.sdsc.edu>. Available at: http://www.gont.com.ar/docs/post-shimomura-usenet.txt.

Silbersack, M. 2005. Improving TCP/IP security through randomization without sacrificing interoperability. EuroBSDCon 2005 Conference.

SinFP. 2006. Net::SinFP - a Perl module to do OS fingerprinting. Available at: http://www.gomor.org/cgi-bin/index.pl?mode=view;page=sinfp

Smart, M., Malan, G., Jahanian, F. 2000. Defeating TCP/IP Stack Fingerprinting. Proceedings of the 9th USENIX Security Symposium, pp. 229-240. Available at: http://www.usenix.org/publications/library/proceedings/sec2000/full_papers/smart/smart_html/index.html

Smith, C., Grundl, P. 2002. Know Your Enemy: Passive Fingerprinting. The Honeynet Project.

Spring, N., Wetherall, D., Ely, D. 2003. Robust Explicit Congestion Notification (ECN) Signaling with Nonces. RFC 3540.

Srisuresh, P., Egevang, K. 2001. Traditional IP Network Address Translator (Traditional NAT). RFC 3022.

Stevens, W. R. 1994. TCP/IP Illustrated, Volume 1: The Protocols. Addison-Wesley Professional Computing Series.

TBIT. 2001. TBIT, the TCP Behavior Inference Tool. Available at: http://www.icir.org/tbit/

Touch, J. 2007. Defending TCP Against Spoofing Attacks. RFC 4953.

US-CERT. 2001. US-CERT Vulnerability Note VU#498440: Multiple TCP/IP implementations may use statistically predictable initial sequence

numbers. Available at: http://www.kb.cert.org/vuls/id/498440

US-CERT. 2003a. US-CERT Vulnerability Note VU#26825: Cisco Secure PIX Firewall TCP Reset Vulnerability. Available at: http://www.kb.cert.org/vuls/id/26825

US-CERT. 2003b. US-CERT Vulnerability Note VU#464113: TCP/IP implementations handle unusual flag combinations inconsistently. Available at: http://www.kb.cert.org/vuls/id/464113

US-CERT. 2004a. US-CERT Vulnerability Note VU#395670: FreeBSD fails to limit number of TCP segments held in reassembly queue. Available at: http://www.kb.cert.org/vuls/id/395670

US-CERT. 2005a. US-CERT Vulnerability Note VU#102014: Optimistic TCP acknowledgements can cause denial of service. Available at: http://www.kb.cert.org/vuls/id/102014

US-CERT. 2005b. US-CERT Vulnerability Note VU#396645: Microsoft Windows vulnerable to DoS via LAND attack. Available at:

http://www.kb.cert.org/vuls/id/396645

US-CERT. 2005c. US-CERT Vulnerability Note VU#637934: TCP does not adequately validate segments before updating timestamp value. Available at: http://www.kb.cert.org/vuls/id/637934

US-CERT. 2005d. US-CERT Vulnerability Note VU#853540: Cisco PIX fails to verify TCP checksum. Available at: http://www.kb.cert.org/vuls/id/853540.

Veysset, F., Courtay, O., Heen, O. 2002. New Tool And Technique For Remote Operating System Fingerprinting. Intranode Research Team.

Watson, P. 2004. Slipping in the Window: TCP Reset Attacks, CanSecWest 2004 Conference.

Welzl, M. 2008. Internet congestion control: evolution and current open issues. CAIA guest talk, Swinburne University, Melbourne, Australia. Available at: http://www.welzl.at/research/publications/caia-jan08.pdf

Wright, G. and W. Stevens. 1994. TCP/IP Illustrated, Volume 2: The Implementation. Addison-Wesley.

Zalewski, M. 2001a. Strange Attractors and TCP/IP Sequence Number Analysis. Available at: http://lcamtuf.coredump.cx/oldtcp/tcpseq.html

Zalewski, M. 2001b. Delivering Signals for Fun and Profit. Available at: http://lcamtuf.coredump.cx/signals.txt

Zalewski, M. 2002. Strange Attractors and TCP/IP Sequence Number Analysis - One Year Later. Available at: http://lcamtuf.coredump.cx/newtcp/

Zalewski, M. 2003a. Windows URG mystery solved! Post to the bugtraq mailing-list. Available at: http://lcamtuf.coredump.cx/p0f-help/p0f/doc/win-memleak.txt

Zalewski, M. 2003b. A new TCP/IP blind data injection technique? Post to the bugtraq mailing-list. Available at: http://lcamtuf.coredump.cx/ipfrag.txt

Zalewski, M. 2006a. p0f passive fingerprinting tool. Available at: http://lcamtuf.coredump.cx/p0f.shtml

Zalewski, M. 2006b. p0f - RST+ signatures. Available at: http://lcamtuf.coredump.cx/p0f-help/p0f/p0fr.fp

Zalewski, M. 2007. 0trace - traceroute on established connections. Post to the bugtraq mailing-list. Available at: http://seclists.org/bugtraq/2007/Jan/0176.html

Zalewski, M. 2008. Museum of broken packets. Available at: http://lcamtuf.coredump.cx/mobp/

Zander, S. 2008. Covert Channels in Computer Networks. Available at: http://caia.swin.edu.au/cv/szander/cc/index.html

Zuquete, A. 2002. Improving the functionality of SYN cookies. 6th IFIP Communications and Multimedia Security Conference (CMS 2002). Available at: http://www.ieeta.pt/~avz/pubs/CMS02.html

Zweig, J., Partridge, C. 1990. TCP Alternate Checksum Options. RFC 1146.

Appendix A. Changes from previous versions of the draft (to be removed by the RFC Editor before publishing this document as an RFC)

A.1. Changes from draft-gont-tcp-security-00

- Draft resubmitted as draft-ietf (boilerplate updated as required).

Appendix B. Advice and guidance to vendors

Vendors are urged to contact CSIRTUK (csirt@cpni.gsi.gov.uk) if they think they may be affected by the issues described in this document. As the lead coordination center for these issues, CPNI is well placed to give advice and guidance as required.

CPNI works extensively with government departments and agencies, commercial organizations and the academic community to research vulnerabilities and potential threats to IT systems especially where they may have an impact on Critical National Infrastructure's (CNI).

Other ways to contact CPNI, plus CPNI's PGP public key, are available at http://www.cpni.gov.uk/ .

Author's Address

Fernando Gont
UK Centre for the Protection of National Infrastructure

Email: fernando@gont.com.ar
URI: http://www.cpni.gov.uk