Detection of malicious Traffic on Backbone Links via Packet Header Analysis

Wolfgang John and Tomas Olovsson
Department of Computer Science and Engineering, Chalmers University of Technology, Göteborg, SE
e-mail: {wolfgang.john, tomas.olovsson}@chalmers.se

Abstract

Purpose: *In this study, modern Internet backbone traffic has been investigated in order to study occurrences of malicious activities and potential security problems within Internet packet headers.*

Design/Methodology/Approach: *Contemporary and highly aggregated backbone data has been analyzed regarding consistency of network and transport layer headers (i.e. IP, TCP, UDP and ICMP). Possible security implications of each anomaly observed are discussed.*

Findings: *A systematic listing of packet header anomalies together with their frequencies as seen "in the wild" is provided. Inconsistencies in protocol headers have been found within almost every aspect analyzed, including incorrect or incomplete series of IP fragments, IP address anomalies and other kinds of header fields not following Internet standards. Internet traffic was shown to contain many erroneous packets; some are the result of software and hardware errors, other the result of intentional and malicious activities.*

Practical Implications: *This study not only presents occurrences of header anomalies as observed in today's Internet traffic, but it also provides detailed discussions about possible causes for the inconsistencies and their security implications for networked devices.*

Originality/Value: *The results of this study are relevant for researchers as well as practitioners, and form valuable input for intrusion detection systems, firewalls and the design of all kinds of networked applications exposed to network attacks.*

Keywords: *Internet Measurement; Security; Header Anomaly; Vulnerability Classification; Malicious Traffic; Backbone Traces;*

Paper Type: *Research Paper*

1. Introduction

In this study, Internet backbone traffic has been investigated with respect to potential security problems and many security-related anomalies in packet headers have been found. Internet traffic contains many erroneous packets; some are the result of software and hardware errors, other the result of intentional and malicious activities. We have searched for anomalies in contemporary, highly aggregated Internet backbone traffic. The results show that header problems can be found within almost every aspect being analyzed. In this study, 27.9 billion frames have been collected and protocol headers on network and transport layers have been analyzed in order to point out all behaviour that could potentially result in a security problem for connected hosts. As a result, a systematic listing of all possible packet header anomalies is provided, including their frequencies as seen "in the wild" on an Internet backbone link. In addition, the possible security implications of each anomaly observed are discussed.

The study of backbone traffic gives a complementary view to studies of traffic with low level of aggregation, such as traffic in local networks. Backbone data provides the opportunity to gain a broader

picture of different types of malicious traffic present on the modern Internet. Besides detection of various types of malicious traffic, specific attack patterns that never show up when studying traffic reaching a smaller network can be observed (e.g. distributed denial of service (DDoS) attacks). Furthermore, rare attacks are more likely to be detected within large amounts of diverse and aggregated traffic and might therefore also reveal previously unreported attack types. Some traffic may seem legitimate when studying only one host but may turn out to be malicious when studying a larger portion of the Internet.

This study sets out to update and extend older studies (Bykova et al., 2001; John and Tafvelin, 2007; Mahoney and Chan, 2001) which have reported about packets not following modern Internet standards (IP, TCP, UDP, ICMP). Additionally, detailed figures about invalid use of fragmentation are included, which is an important security issue previously not covered in the same extent, with the exception of a study about general observations of fragmented traffic (Shannon et al., 2002). While the previous studies only focus on some specific aspects, the present study presents a wider, systematic overview of the most commonly found misbehaviours on the modern Internet and how, and in what extent, common protocols are misused by attackers in their search for vulnerabilities in Internet-connected systems.

As a result, frequencies of occurrence for different kinds of malicious traffic are presented, such as invalid IP packets headers, incorrect use of IP fragmentation, IP address problems, ICMP, UDP and TCP misuse. In the search for anomalies, possible vulnerabilities are listed in table form with references to the packet headers, an approach which should make it easy to find potential problems and make it possible to evaluate the completeness of the study with respect to what header fields are being analyzed. Besides identification of header anomalies deviating from accepted Internet standards, particular well-known attacks and their common names (such as *Land*, *Jolt*, *sPing*, *Teardrop*, *Boink*, etc.) are pointed out in the tables and the analysis.

Many attacks like the ones described above are old and well-known and would therefore be expected to be very rare on the Internet and seemingly unlikely to be found in our data. However, the results of the study show that many of these attacks are still present on the Internet. One reason may be that the recent arrival of new operating systems and mobile devices with small and newly written IP stacks (e.g. mobile phones and PDAs) may have made these attacks meaningful again. A recent example of this was the introduction of Windows Vista, where it turned out that the beta versions were vulnerable to a large number of such old well-known vulnerabilities (Newsham and Hoagland, 2006).

This study shows that it is possible to detect many commonly known attacks from an analysis of network and transport layer packet headers. We therefore believe that our approach to identify potential problems and the scale of our datasets allow us to classify, detect and report a substantial portion of malicious, incorrect and unwanted traffic present on the Internet today. The results of this study should not only be interesting for practitioners and researchers but should also be valuable input for work with intrusion detection systems, firewalls and support the design of all kinds of networked applications that must withstand network attacks.

1.1. Limitations

In our study, it has not always been possible to correlate all packets or to find all series of packets belonging to a malicious activity. There may also be other limitations. The *Smurf* and *Fraggle* attacks, for example, are attacks where ICMP and UDP packets are sent to a network's broadcast address. However, on the Internet backbone there exists no information about what addresses are used for local broadcast messages on smaller networks. Another limitation is that application-level data was removed immediately after data collection making it impossible to inspect the contents of the packets in order to validate, for example, DNS queries or to find problems in application level protocols. Finally, another group of attacks are link-level attacks such as ARP attacks which, of course, are not visible on the Internet and cannot be detected with this type of study. However, we strongly believe that these limitations do not affect the usefulness of this study in any significant way.

1.2. Outline of the paper

Section 2 describes possible anomalies in IP, TCP, UDP and ICMP headers. Different types of anomalies are divided into classes and possible security implications of each type of anomaly are highlighted. In Section 3, the real-life dataset used for the analysis is presented together with the methodology applied. Section 4 provides figures about occurrences of anomalies in the dataset according to the classification scheme presented in Section 2. Besides the figures, each anomaly is discussed and interpreted regarding its security implications. Section 5 concludes the study, highlights the main findings and gives recommendations for how to use the results.

2. Classification of anomalies

In order to provide a systematic overview of possible header misbehaviours and to ease the reading of this paper, the headers of the protocols being analyzed are shown in fig. 1-4 (IP, TCP, UDP and ICMP headers). Header fields highlighted in grey are considered to contain potentially unusual or harmful values and are therefore included in this study. Roman numbers in the figures indicate different types of anomalies within the header fields and the tables below provide descriptions of each anomaly. This type of classification makes it possible to identify all packets violating Internet standards in any way. The motivations for investigating many of the anomalies can be found in the right column of the tables.

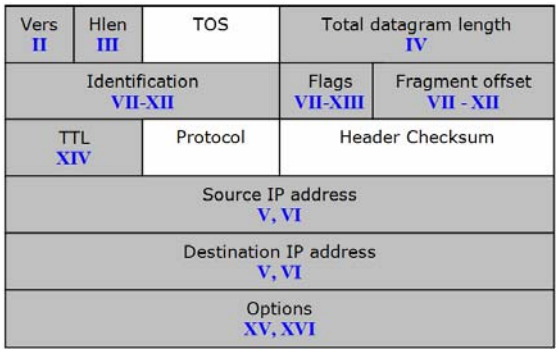

Figure 1: IP header structure

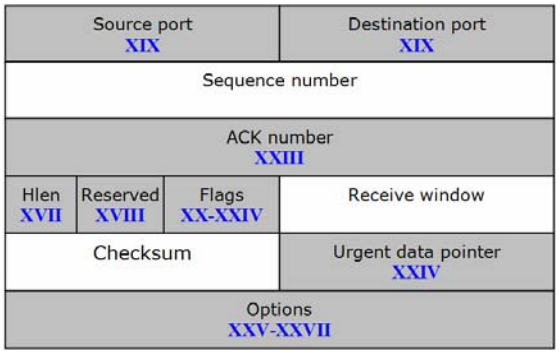

Figure 2: TCP header structure

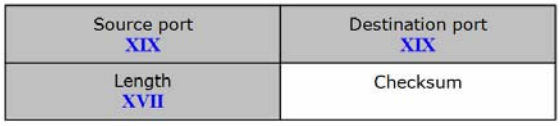

Figure 3: UDP header structure

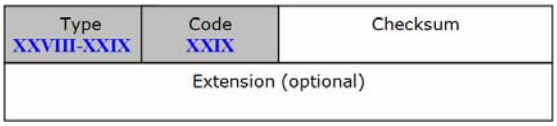

Figure 4: ICMP header structure

General IP header errors:

I	Actual IP packet length is not large enough to host a complete IP and transport header	Truncated packets might be used to confuse firewalls or remote hosts
II	Packet according to HDLC header Ethertype should be IPv4, but IP version is not 4	This is a general protocol error and these packets should have been removed by Internet routers
III	Header length field less than minimum IP header length of 20 bytes	Same type of error as in II above.
IV	Total datagram length value is not sufficient to host IP and transport header (TCP,UDP,ICMP)	Such inconsistencies might be used to confuse firewalls or remote hosts

IP address anomalies:

V	Source address equal to destination address	Can confuse a host to start sending responses to itself (*Land* and other DoS attacks)
VI	Traffic to or from private addresses (RFC 1918) and reserved addresses like loopback, class E, link local or "this network" addresses	These addresses should not be seen. If delivered to hosts, packet may create confusion or may cause unwanted or illegal traffic on local networks.

IP fragmentation anomalies:

VII	First fragment too small to contain full transport header (only for TCP, UDP and ICMP)	No reason such fragments should occur except when trying to confuse firewalls
VIII	Single packets with MF flag or Fragment offset	Either the result of lost fragments or attacker may try to use up buffer space at receiving hosts (DoS)
IX	Gaps in datagram when assembled (including missing first or last fragments)	Attempts to trigger bugs in the reassembly code or to exhaust buffer space at the receiving host (DoS), (*Boink, Opentear, Frag*)
X	Overlapping fragments	Attempts to trigger bugs during datagram reassembly or to traverse traffic filters with malicious code inside the datagram (*Teardrop, Newtear, Jolt, Nestea*)
XI	Duplicate fragments (with or without different contents)	Fragments overwriting its own contents, especially the first fragment, may be used to confuse firewalls that believe they have already inspected the TCP header
XII	Fragment makes assembled datagram exceed max. IP packet length of 64 Kbytes	Attacks where fragment offset plus datagram size exceeds 64 Kbytes IP datagram limit. A possible buffer overflow problem. (*Ping-of-death, sPing, IceNewk*)
XIII	Invalid IP flag combinations	May confuse receiving hosts or firewalls

Potential IP header problems:

XIV	Small TTL values (values smaller 10)	Could be result of topology mapping scan (or legitimately used by e.g. *traceroute*)
XV	IP option(s) used	IP options can be used to circumvent normal routing or to cause other problems (e.g. strict source routing). Not necessarily erroneous, but suspicious.
XVI	IP option length not matching announced IP header length	May confuse receiving hosts or firewalls

General transport header errors:

XVII	TCP header length or UDP length fields less than minimum header length of 20 bytes/8 bytes resp.	May confuse receiving hosts or firewalls
XVIII	TCP reserved bits set	Must be zero according to RFC 793
XIX	Source or destination port of zero (UDP, TCP)	Should not occur in ordinary communication if the host expects a reply. Such packets could confuse hosts when receiving them or replying.

Invalid or unusual use of TCP flags:

XX	Invalid combination of TCP flags (multiple signalling flags, zero flags …)	Examples are *Xmas packets* which are results of setting random flags in hope to create confusion at endpoints or to fingerprint operating systems.
XXI	TCP SYN segment fragmented	SYN segments should never be fragmented
XXII	TCP SYN segment contains data (except T/TCP)	Data in SYN segments serve no practical use
XXIII	ACK number of zero and ACK bit set	Could be result of ACK or FIN scan attacks.
XXIV	Urgent data pointer value when URG flag set	Has been used for DoS attacks (e.g. *WinNuke*)

TCP option errors:

XXV	TCP option type invalid	May confuse receiving hosts or firewalls
XXVI	TCP option length not matching header length	May confuse receiving hosts or firewalls
XXVII	TCP option length equal to zero	Applications (e.g. Symantec Personal Firewall) might loop endlessly when parsing such options

ICMP anomalies and general statistics:

XXVIII	ICMP length anomalies	ICMP messages not following standards (too small or too large for specific type/code)
XXIX	ICMP types and codes	Source Quench may slow down senders (DoS), Redirect may place attacker as man in-the-middle

3. Data description and methodology

The dataset used in this study (John and Tafvelin, 2006) was collected from September to November 2006 on an OC192 backbone link of the Swedish University Network (SUNET). The packet header traces have been collected on a highly aggregated backbone link at 277 randomly selected times during 80 days, in order to provide a good statistical representation of all Internet traffic during the time-period at this location. At each randomly selected time, two traces of 10 minutes duration were stored. When recording the packet level traces on the 2x10GB links, payload beyond transport layer was removed and IP addresses were anonymized due to privacy concerns using the prefix preserving CryptoPAN (Xu et al., 2001). After

further pre-processing of the traces as described in (John and Tafvelin, 2006) and (John and Tafvelin, 2007), the traces were moved to a central storage. An analysis program was run on the raw traces to extract malformed packet headers and invalid series of fragments. The reduced data was then stored in a database together with statistical summaries for each particular observation as listed in Section 2. For packets of special interest, corresponding flows have been extracted from the raw traces and analyzed in detail using available packet visualization software.

The complete dataset consists of 554 traces including 27.9 billion frames. 99.98% of the traffic was IPv4 carrying 19.5 TB of data in 636 million flows. During the single 10 minute intervals, depending on time of day, between 13,000 and 37,000 unique IP addresses were observed belonging to the region of western Sweden connecting to 300,000-1,000,000 unique addresses on the main Internet. A breakdown of transport protocols in the IP traffic is summarized in Table 1. Not only numbers and fractions of packets are shown but also IP fragments and fragment series are listed (first three lines).

	IPv4	TCP	UDP	ICMP	GRE	ESP
Packets total	27,873,847,645	89.7%	9.8%	0.3%	0.1%	0.1%
Fragments total	255,470,635	0.2%	99.3%	0.0%	0.0%	0.4%
Frag. Series total	20,752,539	1.5%	95.7%	0.0%	0.1%	2.7%
Fragments w/o bulk transfer	42,755,210	1.5%	95.7%	0.0%	0.1%	2.6%
Frag. Series w/o bulk transfer	11,337,769	2.7%	92.1%	0.0%	0.2%	5.0%

Table 1: Transport protocol breakdown

The analysis of fragmented traffic requires correlation of fragmented IP packets to create fragment series. Following Shannon (Shannon et al., 2002), a fragment series has been defined as a list of fragments observed on the network derived from a single original IP packet. Consequently, fragments have been grouped into fragment series based on the IP ID, protocol and the source and destination IP fields. Furthermore a timeout value of one second was chosen to further separate fragment series. As opposed to the study by Shannon et al. (2002) which was carried out on OC12 links, the timeout had to be chosen smaller in order to compensate for the higher throughput of the links measured (OC192). In rare cases, wraparounds of the IP ID space have been observed within a few seconds, which made such a small timeout necessary. Considering the transmission rates of modern computers, a timeout of one second seems to be sufficient to capture all fragments belonging to a certain series, which was proven to be true by a number of empirical tests on the dataset. Furthermore, fragment series observed in the first or the last second of a measurement interval have not been tested for completeness, in order to avoid bias due to border effects.

Earlier studies have shown that only 0.06% of the traffic was fragmented on the measured network (John and Tafvelin, 2007). The increased fraction of IP fragments in the dataset used for this study (0.9%) is explained by a special bulk data transfer event from a space observatory to a data centre in Europe. During 7 time intervals, 213 million fragments in 9.4 million fragment series have been transferred on the outgoing link using a customized fast bulk transfer protocol based on UDP. Figures for the remaining fragmented traffic without the mentioned bulk transfer are summarized in the final two rows of Table 1. Disregarding this special event, only about 0.15% of the IPv4 traffic was fragmented.

4. Observed misbehaviours and anomalies

In the following subsections, occurrences of the anomalies classified in Section 2 as seen in our dataset, are presented in tables. The index columns use the same Roman numbers as introduced earlier. For IP level anomalies a transport protocol breakdown is provided as well. Each table is followed by remarks and

discussions about the anomalies being observed, including an interpretation and discussion of their probable causes.

4.1. IP header anomalies

I – IV: Packets with an insufficient actual packet length to carry the minimal IP and transport headers (I) have been seen very rarely, originating from different IP addresses at different times. 105 of these packets also announced insufficient sizes in the IP total length field (IV). IP version numbers not agreeing with the HDLC Ethertype (II) or IPv4 header length fields smaller than 20 (III) have not been seen at all. Since these errors rarely happen and no well-known attacks exploit such anomalies, we believe these packets are caused by rare IP stack errors.

V: Packets with a source IP address equal to the destination address, as used in the *Land* attack, have been seen 321 times. The original land attack was based on TCP SYN packets which has been observed 9 times in the dataset. Most packets with this anomaly are UDP segments, which means that they are modified versions of *Land*. These packets have been observed at 158 different times, sent between a number of different IP addresses.

VI: IP packets to or from reserved address spaces have been observed in relatively large numbers. A couple of hundred such packets are observed at each of the 277 measurement times, with exception of two 10 minute intervals with peak numbers of around one million each. The majority (95%) of these packets use a source IP addresses belonging to the private class C address room 192.168 /16 even if private class A (10 /8) and class B (172.16 /16 – 172.31 /16) have also been observed as source addresses (5%). Traffic from loopback, link-local, class E or this-network addresses have been recorded, but in very low numbers. Most of the packets in this category are ICMP echo replies (type 0) with length of 228 bytes to four destination hosts during two measurement intervals. We believe that this was an ICMP DoS attack, where Echo replies were chosen to evade stateless firewalls. In order to disguise the real origin, spoofed private addresses were chosen. The remaining 300,000 packets, which appear in a more random and spread out fashion, could also be attacks but might as well be caused by misbehaving or misconfigured NAT gateways.

Index	# packets	TCP	UDP	ICMP	Description
I	123	104	11	8	Insufficient actual packet length
II,III	0	0	0	0	IP version and IP header length fields
IV	105	102	0	3	IP total datagram length field
V	321	9	309	3	source IP addr. = destination IP addr.
VI	2,663,891	185,863	33,780	2,444,232	Reserved address space
XIII	265,324	42,632	222,667	4	Invalid IP flags
XIV	8,067,930	896,790	1,915,931	5,199,576	Small TTL values (<10)
XV, XVI	21,991	0	18,721	2,318	IP options

Table 2: Packet counts observed in 27.8 billion IP packets

XIII: The only defined values for the three IP flags are don't fragment (DF), more fragments (MF) or no bits set. However, 265,000 packets with other, undefined bit values have been observed, which is an increase compared to previous studies (John and Tafvelin, 2007). All possible bit combinations have been seen, with MF+DF responsible for 99% of the invalid combinations. Most IP packets with invalid flag values carry UDP traffic, but no source or destination hosts or port numbers stand out. Furthermore, such packets are seen within each of the 277 traces, with a few traces carrying relatively large numbers of up to 10% of the packets. We believe that these packets are mainly forged packets by hacker tools like *nmap* in order to test robustness of implementations. Furthermore, in a previous study (John and Tafvelin, 2007) also erroneous IP stack implementations have been found to contribute to this behaviour.

XIV: While the usage of small TTL values is no unusual behaviour per se, it might still indicate topology map scanning as preparations for specific attacks towards a network. Modern operating systems use default TTL values of 60 or more. Network paths with hop-counts of more than 50 are very rare, which means that IP packets with small TTL values can be explained by:

- old Windows systems with TTL values of 32 (hop-counts between 20 and 30 are plausible)
- packets from *traceroute* applications (commonly only ICMP and UDP)
- topology mapping scans using TCP or UDP on common ports in order to avoid ICMP filters

Indeed, 99.6% of the ICMP packets with a small TTL are of type 8 (echo), which is used for *traceroute* in Windows systems and some Unix versions. The remaining ICMP packets with small TTL values are of type 3 (destination unreachable). Their packet length of 28 bytes indicates that they are replies to either UDP or ICMP packets. Hosts receiving these ICMP type 3 messages typically show heavy activity on UDP ports known to be commonly used for P2P signalling traffic, which leads to the conclusion that the messages are artefacts of the unreliable nature of P2P overlay networks and thus are most likely not a security issue.

Many Unix systems use UDP packets with varying size of around 40 bytes for *traceroute* with a destination port within the otherwise uncommon port range 33434 to (around) 33534. 92% of the UDP packets with short TTL fall into this port range and since they are small in packet size, they are most likely legitimate *traceroute* packets. The remaining 8% UDP packets are not only directed to other random port numbers, they also have larger packet sizes between 100 and 1500 bytes, which indicates that they must be treated as suspicious.

Most of the TCP packets with small TTL values are the downstream part of regular TCP connections. They also show TTL values of 8 or 9 which is larger than the *traceroute* traffic observed for ICMP and UDP (TTL values of 3, 5 or 6 for this specific topology), which consequently is too large for topology mapping scans. These remote hosts are therefore most likely running a system with a small default TTL value, such as Windows NT or 95. This leaves about 5300 packets to be suspected as topology mapping scans via TCP (suitable TTL values with small packets).

Additional 55,500 packets of protocol type 103 (PIM–protocol independent multicast) have been observed with TTL values of exactly one – which are valid PIM bootstrapping messages following RFC 2362.

XV, XVI: As already observed in a previous study (John and Tafvelin, 2007), IP options are rarely used. Source routing is the main security concern regarding IP options, but has not been observed at all (neither IP option value 131 nor 137). A large part of the packets with IP options is sent by 10 sources to one destination inside Gothenburg via UDP. Strangely, instead of using real options, a sequence of four EOOL bytes is sent (0,0,0,0) which is most likely due to inappropriate configuration or buggy software. The options used by ICMP traffic are of valid option type 7 (record route), and the remaining 948 IP options observed are option type 148 (router alert) being sent within RSVP packets. IP option header length inconsistencies (XVI) have not been observed.

4.2.IP fragmentation anomalies

The figures of IP fragmentation anomalies in the dataset used are skewed due to one exceptional event, where exactly one host inside a University was sending UDP segments fragmented into 6-7 fragments to five different hosts at five different measurement times during a few days, in very high frequency. As destination port number, the entire 16 bit port space was used in iterative fashion. About 50% of the IP series included different types of inconsistencies, most with missing last fragments, but also other gaps and a number of “single packet series”. Most likely this is a hijacked host used for directed DoS attacks (such as a *Frag attack*). The 1.6 million fragment series from this host have been summarized in the first row of Table 3 and are excluded from the remaining analysis.

Index	#series	TCP	UDP	ICMP	Description
VII-IX	1,651,324	0	1,651,324	0	Exceptional fragmentation event
VII, XXI	71	71	0	0	Series with short first fragment
VIII	80,981	18,117	61,001	1,723	Single packet "series"
IX	29,939	685	29,217	37	Incomplete series ("gaps")
X	37	5	32	0	Series with overlapping fragments
XI	1,864	1,285	579	0	Series with duplicated fragments
XII	0	0	0	0	Series exceeding 64K IP length

Table 3: Fragment series counts observed in 20.8 million series

VII, XXI: Fragment series where the first fragments are too small to contain all headers (VII) are observed in 71 series sent by a single host. The fragments had furthermore the TCP SYN flag set (XX) and the IP total length field was smaller than 40 bytes (IV). This is probably a DoS attack trying to confuse firewalls or receivers.

VIII: About 81,000 packets appear to be part of a fragment series (either MF bit or fragment offset set), but no other fragments are observed for the same series. This could potentially be used to confuse hosts or firewalls. Another plausible explanation is that further fragments have been dropped or routed asymmetrically. Around 10% of the single-packet series used IP IDs of zero, which means that IDs of zero are around 1,500 times more common than any other possible number in the 16-bit space. This obvious over-representation of one IP ID is suspicious and indicates either malicious intentions or usage of protocol implementations not following Internet standards. 58% of the single fragment series have the MF bit set and are full sized packets (~1500 bytes), looking like typical first packets in fragment series. The remaining 42% have characteristic properties of last packets in fragment series, with IP offset values set and small data portions. These observations suggest that dropped packets could be an explanation for many of these packets.

IX: Incomplete fragment series are very undesirable because they consume resources at the receiving host which needs to store the arriving fragments until the series is complete and the entire packet can be handed over to the next protocol layer. Known attack types in this category are *Opentear* and *Frag*. Around 50% of incomplete fragment series are missing the last fragment, a missing first fragment accounts for 25% and the remaining 25% are gaps in between. In the dataset, 42 hosts receive about 80% of all the incomplete series. The incomplete series are sent by different hosts and are targeted to random UDP ports. The sizes of the gaps range from 8(!) bytes to full packet size. Besides these incomplete series, valid series of fragments are also sent to the hosts in question and only around 1/5 of all fragment series are actually incomplete. Even though this behaviour could be explained by a high number of packet losses along the path, a suspiciously high density of IP IDs of zero together with the unusual gap size of 8 bytes makes it more likely that these hosts are the target of a DDoS attack by a number of bots, similar to the hijacked host causing the exceptional event described above.

X: Overlapping fragments are also known to be a common DoS attack type (such as *Teardrop, Jolt* and *Nestea*). Overlaps are very rare in the dataset and only 37 occurrences have been observed at 35 times between different hosts, mainly UDP. Almost all series with overlaps (35) also include missing sections in the complete IP datagram. The small overlapping fragments (8 to 48 bytes) have exactly the same size as the gaps in the specific series, but they fill the gaps at the wrong offset. Depending on the length of the series, such overlapping fragments appear up to 3 times per series, with the last byte of the overlapping fragment always at the datagram offset of 912, 1832 and 5352 bytes. We believe that this consistent behaviour is either the result of a soft- or hardware error or an attack tool repeating the same behaviour.

XI: In contrast to overlaps, duplicate fragments mean that two fragments cover the exact same portion (offset and fragment length) within the fragmented datagram. Potentially, this could cause similar problems as overlaps in *Teardrop* attacks, namely overwriting previous benign portions with new malicious data. Since packet payload in our dataset has been removed, we need to rely on transport header checksum

information (note that transport header checksums in fragment series are only available in the first fragment, so changed payload in following fragments cannot be detected in the present study). According to a checksum comparison of duplicate first fragments, many duplicate fragments observed are in fact sheer retransmissions. There are also sequences of duplicated single fragment series (VIII) which are therefore categorized as duplicates as well. Note that an unproportional large number of these single series duplicates use IP IDs of zero, which again appears to be a good criteria for identification of malicious fragmented traffic. Duplicated fragments with different payload (and consequently different transport header checksums) within otherwise complete and valid series have only been observed 104 times by 21 hosts.

XII: Attacks, with fragment series exceeding the maximum IP packet length of 64 Kbytes (*Ping-of-death*, *sPing*, *IceNewk*), are not observed at all. This attack type was popular in the late 90's, but since then most applications and operating systems have been patched. Even though it is good news that this attack is not observed anymore, application developers and firewall administrators should keep this attack in mind.

4.3.TCP header anomalies

In the first row of Table 4 TCP segments with multiple invalid header fields have been summarized. Garbled TCP headers have been defined as combinations of two or more independent[1] field anomalies within one TCP header. Garbled headers have been observed during all measurement intervals with no specific host standing out. Such packets can easily be forged by network exploration tools using raw sockets, such as *nmap*. Note that the segment counts in the following categories do not include the 9,757 garbled TCP headers.

Index	# segments	Description
XVII - XXVII	9,757	Garbled TCP header
XVII	72	TCP length short
XVIII	114,876	Reserved bits set
XIX	6,180	TCP port zero
XX a	178,993	Invalid signaling flags
XX b	81,982	pure FIN (no ACK)
XXII	29,369	SYN with data
XXIII	389,060	ACK number of zero
XXIV	440	Urgent pointer set
XXV - XXVII	9,038	TCP option errors

Table 4: TCP segment counts observed in 25 billion TCP segments

XVII: TCP headers with length values smaller than the minimum TCP header length of 20 bytes have been observed 72 times. The announced header length values are mainly 0 and 8 bytes.

XVIII: The reserved bits in the TCP header are the four bits following the TCP header length field. The previously six reserved bits have been reduced to four since the introduction of ECN (RFC 3168). In this study, the ECN bits have not been considered. According to the TCP specification (RFC 793) the reserved bits must be zero. However, almost 115,000 packets with non-zero reserved bits have been observed in the dataset. Interestingly, TCP reserved bits are only set together with TCP flag combinations of either RST/ACK or SYN/ACK. 21% of the invalid reserved bits have all 4 bits set and appear consistently with RST/ACK packets. These RST/ACK packets are mainly replies from HTTP servers on port 80. Valid TCP conversations are closed by the servers with sequences of 3-4 RST/ACK packets, where each but the initial

[1] i.e. anomalies in different fields. XVIII / XX and XXV-XXVII are considered dependent

RST/ACK has all reserved bits set. This appears to be an incorrect TCP implementation rather than a security issue.

The remaining 79% of TCP headers with invalid reserved bits appear within SYN/ACK packets only, but this time with different bit combinations. These SYN/ACKs are sent in high frequency by different sources, usually from port 80 or 7000. Interestingly, no SYN packets triggering these SYN/ACK responses and no further packets originating from these sources have been seen. This behaviour can either be explained by an asymmetrically routed (and therefore un-captured) SYN attack, but more likely are SYN/ACK attacks. In this case the often used source port of 80 can be explained by attempts to pass certain stateless firewalls.

XIX: TCP port number zero is reserved and should not be used for data transfer. Nevertheless, 6,180 TCP segments with a port number of zero have been observed equally shared between source and destination port numbers. These segments have been sent by approximately 700 different sources within almost all measurement intervals. Most of these packets are SYN packets being part of host scanning campaigns. Very few of them are actually replied to with RST/ACK packets.

XX: Anomalies within the 6-bit TCP flags field have been divided into invalid combinations of the signalling flags SYN, FIN and RST (XXa) and another, less critical, but still unexpected flag value (XXb).

XX a: Combinations of invalid signalling flags appear in frequencies comparable to observations in an earlier study (John and Tafvelin, 2007). 826 segments had no signalling flags set at all (zero flags). The combination RST+FIN in the same header has been observed 939 times, SYN+FIN 435 times and 377 segments had SYN+FIN+RST set. The most common flag anomaly however is RST+FIN with more than 176,000 occurrences. Packets with invalid signalling flag combinations have been seen evenly distributed within all traces, sent by more than 45,000 hosts to different destinations where approximately 50% are directed to port 80. The most likely explanation for such segments is crafted packets (such as *X-mas*) by network exploration and testing tools like *nmap*.

XX b: Beside combinations of signalling flags, another type of unexpected flag values has been observed. According to the TCP specification, every segment in an established TCP connection (except the initial SYN) is required to carry an ACK, so there is no reason for pure FIN packets to exist. Mahoney (Mahoney and Chan, 2001) showed that identification of FIN packets without an ACK can reveal port-sweeps and OS fingerprinting campaigns. In our dataset, 82,000 segments with only FIN flags set have been observed, sent by 10,000 different hosts to 27,000 destinations. Interestingly, more than 50% of these packets are sent to different well known P2P ports. Most pure FIN packets are sent after a sequence of SYN connection attempts just before the socket is finally closed on the sending host after the TCP timeout. Even if this behaviour is not defined in the standards, we do not consider it to be security relevant.

XXII: According to RFC 793, it is not prohibited to append payload data to SYN packets. However, this behaviour is de facto non standard and therefore somewhat suspicious. The only well defined usage of SYN packets with data is T/TCP (RFC 1644) which can be identified by TCP options. However, this has not been observed in this dataset. Around 29,000 SYN packets with data had a data portion of exactly 24 bytes and have been sent to TCP port 53 (DNS). These packets are seen quite evenly distributed among all measurement times and are exchanged between about 113 different IP addresses outside the region of Gothenburg to 66 hosts inside. The connections are initialized by this SYN data segment, replied by a SYN/ACK and then immediately closed by the initiator with a RST. According to SANS Intrusion Detection FAQ (SANS, 2008) this behaviour has been observed in other networks and is probably caused by a common but buggy DNS system. Other segments with SYN flags and data have only been observed in packets with garbled TCP headers.

XXIII: TCP sequence and acknowledgement numbers use 32-bit integers. Even if the selection of initial sequence numbers (ISN) is known not to be completely random for many systems (Zalewski, 2002), 390,000 out of 25 billion segments having an ACK number of zero is a clear overrepresentation. It turned out that 96% of these packets are RST/ACK segments. A large portion of these segments is sent by hosts

closing valid connections with series of RST/ACK packets. In these connections, all but the initial RST/ACK packets carried an ACK number of zero, which appears to be an implementation problem rather than malicious activity. In addition, some RST/ACK storms have been observed with no SYN packets that could have triggered these replies. This behaviour could be explained by an asymmetrically routed (and therefore un-captured) SYN attack, but more likely as RST/ACK attacks where the ACK number field was left empty (zeros). The remaining ACK numbers of zero (4%) are pure ACK packets sent between a large number of hosts, thus there is no indication of obvious malicious intentions.

XXIV: Urgent pointers are basically a valid way to transmit "out of band" data within the regular TCP stream of a connection, used e.g. for quick delivery of control strings in applications like *telnet* or *ftp*. However, in the past urgent pointers have turned out to be an effective way for DoS attacks due to buggy operating systems (*WinNuke*) - regardless of the actual value of the urgent pointer. This means that many firewalls today drop all packets with urgent pointer flags. In the dataset, 3,389 TCP segments carried an URG flag, though none of these packets were directed to port 139, the target of the infamous original *WinNuke* attack. Most of the URG segments had generally garbled TCP headers and only 440 "pure" URG flags have been observed. 71 of these packets have been sent to port 21 (*ftp*) with plausible urgent pointer values of 2 – 4 (e.g. for control characters like "ctrl-c"). The remaining pure URG segments have been sent to different P2P port numbers with urgent pointer values of one or zero. Especially urgent pointers pointing to a data offset of zero are suspicious since it indicates that there is in fact no data to deliver urgently.

XXV - XXVII: TCP option anomalies in this dataset have been observed in frequencies comparable to results of a previous study (John and Tafvelin, 2007). Three different anomalies (XXV-XXVII) have been observed in different combinations within 9,000 TCP segments. Such packets are most likely crafted by tools like *nmap*. The most common inconsistencies are either usage of undefined option types (TCP options are only defined for type numbers up to 26) and length announcements in the length field of specific options which do not agree with the header length of the general TCP header. Additionally, 967 options carried an option length value of zero which has been shown to cause endless loops when processing them in receivers and traffic filters (e.g. Symantec Personal Firewall).

4.4.UDP header anomalies

Index	# packets	Description
XVII	67	UDP length field
XIX	17,242	UDP port zero

Table 5: UDP segments observed in 2.7 billion UDP segments

XVII: 67 packets with too small values in the UDP length field have been observed in 56 different measurement intervals sent by 59 different hosts. The most common invalid header length value is zero. Only two packets announced header length of one and two, which is too small to carry the minimum UDP header length of 8 bytes. The low frequency of this anomaly does not allow us to reliably classify this as a malicious action.

XIX: As for TCP, UDP port numbers of zero are also reserved. In the dataset, around 3,000 UDP segments have been sent to UDP port zero, which is definitely not permitted and can lead to crashes of hosts or firewalls. According to the UDP specification (RFC 768) the source port number field is optional and may be set to zero if not used, i.e. no reply is expected. A large portion of about 14,000 packets has been seen with source port values of zero. Even if this behaviour per se is permissible, it turned out that all the segments coming from UDP port zero are sent in short scanning campaigns, scanning over ranges of /24 networks (254 IP addresses) on port numbers 1025 and 1026 (win-rpc). Such campaigns have been launched by 30 different hosts at 30 different times. The payload length of all these packets was consistently 319 bytes. UDP Source port numbers of zero therefore seem to be good indication of windows

messenger spam, where spammers sweep over IP ranges and try to deliver pop-up messages to windows systems with windows messenger active.

4.5.ICMP anomalies and observations

XXVIII, XXIX: A breakdown of observed ICMP packet types is presented in Table 6. ICMP messages with undefined type or code are summarized in the second last row of the table. Furthermore, messages with impossible length values according to their ICMP types are summarized in the last row. This means that the counts and fractions presented per ICMP type are counts of packets with valid types and codes and plausible length values. In the following paragraphs, this table will be analyzed regarding possible ICMP attacks.

Ping-of-death type attacks, where a fragmented ICMP packets exceed 64 Kbytes when assembled, have not been observed (in fact no such fragments attacks were detected, see XII). There where also no ICMP *p-Smash* attacks (floods of **ICMP router advertisements** (ICMP type 9)).

Spoofed **ICMP destination unreachable** messages (type 3), as used in a *Smack* attacks, could be present in the dataset, but are difficult to pinpoint in this study due to anonymized IP addresses and missing payload of the ICMP messages. However, neither source nor destination hosts appeared to be involved in unusually dense sequences of ICMP type 3 messages during the 277 measurement times.

Also **ICMP source quench messages** (type 4) have been reported to be exploited in order to slow down networks. In the dataset, almost 38,000 such messages have been seen and even if such DoS attacks cannot be ruled out, no obvious attack patterns were identified.

ICMP type	# packets	Percent	Description
0	5,927,990	6.10%	Echo Reply
3	11,964,456	12.31%	Destination unreachable
4	37,899	0.04%	Source Quench
5	46,437,420	47.77%	Redirect
6	1	0.00%	Alternate Host Address
8	16,287,609	16.76%	Echo
11	10,160,608	10.45%	Time Exceeded
12	60	0.00%	Parameter Problem
13	63	0.00%	Timestamp
14	60	0.00%	Timestamp Reply
15	2	0.00%	Information Request
17	10	0.00%	Address Mask Request
Undefined	33,517	0.03%	Undefined ICMP type or code
Invalid length	9,467,433	9.74%	Valid type and code, but invalid length

Table 6: Breakdown of 97.2 million ICMP packets

The large number of **ICMP redirects** is caused by two hosts sending about 46 million ICMP packets with type 5, code 1 (host redirect) to 300,000 destinations during the measurement intervals within a period of 12 days. The general behaviour of these hosts clearly shows that they are not routers or gateways but rather normal workstations establishing only connections to HTTP and P2P hosts. Most likely, these packets are part of a DoS attack like *Winfreez*, which can cause windows machines to change their routing tables. Unfortunately, missing packet payload makes it impossible to analyze the announced gateway addresses in the redirect messages observed.

ICMP timestamp attacks, like *Moyari13*, cannot be identified since the timestamp information has not been preserved in the dataset. However, all timestamp messages (type 13 and 14) have valid packet

lengths. Furthermore, all except three timestamp messages (type 13) have immediately been replied to (type 14), which does not indicate malicious behaviour.

Undefined ICMP types and codes could potentially be part of *Twinge* or *Trash* attacks, which cycle through all types and codes, thereby trying to create confusion or crash certain operating systems. The 33,517 packets with random types and codes in the dataset have been sent between a couple of thousand hosts quite evenly distributed among all trace intervals, with no host or time interval standing out, which means that at least large scale campaigns of these attacks have not been observed.

Finally, a quite large number of packets with valid types but **invalid packet lengths** have been observed. According to RFC 792 (ICMP), most messages except echo have well defined packet sizes or are at least bound to a maximum (often 56 bytes including 20 bytes IP header, 8 bytes ICMP header, 20 bytes original IP header and up to 8 bytes of original payload). Almost all ICMP packets with invalid lengths are of type 3 (destination unreachable) and the remaining 2% are of type 11 (time exceeded), having packet sizes exceeding 56 bytes. Since a large number of hosts were sending these packets in small frequencies, it is likely that most of these are the result of implementations with wrong interpretation of the ICMP standard. Only three ICMP messages have been observed with insufficient IP packet lengths to host an ICMP header.

5. Summary and Conclusions

In this paper, we first provide a systematic classification of header fields not following Internet specifications with a potential to cause security problems. This systematic classification serves as a starting point for identification of such header inconsistencies within our large dataset consisting of packet header data collected on a contemporary, highly aggregated Internet backbone link with diverse traffic composition. Occurrences of each header anomaly as observed “in the wild” are then presented followed by detailed discussions about possible causes and an interpretation of the observations with respect to security relevance.

As a general observation, it is surprising to see that that many old, well known attacks can still be found. On the upside, some former popular attacks, such as *Ping-of-death* and the *IP source route exploit* have not been observed at all. Generally, a constant “noise” of malformed or inconsistent packet headers was observed, similar and consistent with the observations of constant scanning activities in another recent connection-level study (John et al., 2008). This type of background noise is in some cases likely to be caused by rare hardware and software errors, but most must be attributed to the possibilities even inexperienced hackers have today to generate more or less random packet headers with existing networking tools.

Also a number of exceptional events of malicious activity have been observed. An ICMP DoS attack with otherwise unsuspicious echo reply messages has been identified due to IP address analysis regarding reserved IP spaces. A sequence of fragmented datagrams has been sent in high intensity from a single host during short time intervals. The detailed analysis of the fragment series revealed a directed *Frag* attack, using incomplete fragment series with the intention to exhaust resources at the receivers. Furthermore, an analysis of IP ID values of zero appeared to be a successful approach to detect different fragmentation anomalies, and observations in the reserved bits field of the TCP header revealed a series of SYN/ACK attacks. Port number values of zero proved to be effective in detecting port scanning campaigns, both for TCP and for UDP. Finally, a DoS attack applying ICMP redirect messages has been observed.

There are many interesting future research possibilities to improve the results and insights of this study, such as a complementary flow-level investigation of scanning traffic or a similar packet inspection including at least some application-level data, to name but a few. However, the results of this study show that it is possible to detect a substantial part of malicious activities just from inspection of header data. The observations also show that inspection of IP addresses spaces, IP ID values, port number values, the entire flags section in the TCP header (reserved bits and signalling flags) and ICMP messages are the most

effective mechanisms to find malicious traffic and therefore form a basic set of rules which should be included into all modern firewall and IDS systems.

The results presented here are based on a rare dataset of aggregated backbone traffic and are intended to guide and support network administrators and application developers in their constant task of tuning their systems in order to mitigate the wide range of incoming malicious attack traffic. Furthermore, we believe that this study helps researchers and practitioners to gain a better understanding of the characteristics of today's Internet traffic in order to remain proactive.

Acknowledgements

This work was supported by SUNET, the Swedish University Computer Network. The authors furthermore want to thank Prof. Sven Tafvelin for valuable discussions and comments.

References

Bykova M. and Ostermann S. and Tjaden B. (2001), "Detecting Network Intrusions via a statistical Analysis of Network Packet Characteristics", Proceedings of the 33rd Southeastern Symposium on System Theory, pp. 309-314

John, W. and Tafvelin, S. (2006), "SUNET OC 192 Traces, fall 2006", DatCat, available at: http://imdc.datcat.org/collection/1-04HQ-3=SUNET+OC+192+Traces\%2C+fall+2006 (accessed 22 July 2008)

John, W. and Tafvelin, S. (2007), "Analysis of Internet Backbone Traffic and Header Anomalies observed", Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement, ACM, New York, NY, pp. 111-116

John W. and Tafvelin S. and Olovsson T. (2008), "Trends and Differences in Connection Behavior within Classes of Internet Backbone Traffic", in Claypool M. and Uhlig S. (Eds.), Proceedings of the 9th Passive and Active Measurement Conference, Springer, Berlin, pp. 192-201

Mahoney M. and Chan P. (2001), "PHAD: Packet Header Anomaly Detection for identifying hostile Network Traffic", Florida Tech, Technical Report CS-2001-4

Newsham T. and Hoagland J. (2006), "Windows Vista Network Attack Surface Analysis: A Broad Overview", Symantec Corporation, available at: http://www.symantec.com/avcenter/reference/Vista_Network_Attack_Surface_RTM.pdf (accessed 22 July 2008)

SANS Institute (2008), "SANS Intrusion Detection FAQ", available at: www.sans.org/resources/idfaq/dns.php (accessed 22 July 2008)

Shannon C. and Moore D. and Claffy K.C. (2002), "Beyond Folklore: Observations on fragmented Traffic", IEEE/ACM Transactions on Networking, Vol. 10, No. 6,, pp. 709-720

Xu J. and Fan J. and Ammar M. and Moon S.B. (2001), "On the Design and Performance of Prefix-preserving IP Traffic Trace Anonymization", Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement, ACM, New York, NY, pp. 263-266

Zalewski M. (2002), "Strange Attractors and TCP/IP Sequence Number Analysis - One Year Later", available at: http://lcamtuf.coredump.cx/newtcp/ (accessed 22 July 2008)