Statistical Analysis of Malformed Packets and Their Origins in the Modern Internet

Marina Bykova, Shawn Ostermann
School of Electrical Engineering and Computer Science
Ohio University
mbykova@irg.cs.ohiou.edu, ostermann@cs.ohiou.edu

Abstract— **In this work, we collect and analyze all of the IP and TCP headers of packets seen on a network that either violate existing standards or should not appear in modern internets. Our goal is to determine the reason that these packets appear on the network and evaluate what proportion of such packets could cause actual damage. Thus, we examine and divide the unusual packets obtained during our experiments into several categories based on their type and possible cause and show the results.**

I. Introduction

It is well known that some protocol implementations and network applications are not fully conformant with the standards and might "pollute" a network with incorrectly formed packets. A number of common implementation problems were even documented, for example, in [17]. Intentional abuse of network resources can also become a threat to sound communication for the communicating hosts as well as other machines that rely on services provided by the global Internet.

Since control information that is necessary for the correct performance of network protocols is carried in their packet headers, it becomes possible to detect abnormalities in packets seen on a network by checking their headers. In this work, we collect and analyze all of the IP and TCP headers of packets seen on a network that either violate existing standards or should not appear in modern internets. Some work has been perform in the same area to either use malformed packets in network intrusion detection [12] or normalize traffic by adjusting packet header fields to reduce damage caused by such packets [8]. We do not try to identify all network attacks or build a system to defeat against malformed packets, but rather try to determine how much information about the origin and possible cause we might be able to obtain by looking at packet headers while ignoring their contents. The questions we try to answer here are as follows:

- *What is the reason that these packets appear on the network?*
- *How often do we see them?*
- *What proportion of such packets could cause actual damage?*

In section II we present a description of the experiment. Section III covers our results and section IV summarizes statistical results and and give various distributions of error rates. Lastly, in section V we summarize our findings.

II. Description of the Experiment

A. Link Description

In order to perform this research, we used data captured from two different sources. For the first source, we monitored Ohio University's main Internet link, the only link in and out of the university. These traces were obtained from a 100Mbps Fast Ethernet connection rate-limited to 36Mbps between Ohio University (OU) and its ISP and carry packets for approximately 20,000 local hosts.

The second source of data was a 10Mbps Ethernet LAN carrying OU student dormitory traffic. There were approximately 2,500 computers connected to this network.

B. Tools Used

We used `tcpdump` [10] to capture data from the monitored links and `tcptrace` [15] in real-time mode to analyze it. We modified `tcptrace` for the purpose of this analysis and wrote a special module for it. Further analysis and calculation of statistical results were performed with Perl and shell scripts.

C. Packet Analysis

Our analysis is based on the IP and TCP headers of packets from the monitored traffic[1]. We perform an IP header analysis of all of the IP packets regardless of their transport layer protocol and analyze the headers of all TCP packets. UDP, ICMP and other types of packets could also be used in security breaches but their analysis is out of scope for this work. This section describes all of the packet header fields that we included in the analysis and illustrates what values we considered abnormal.

C.1 IP Header Analysis

1. **Packet Size**. We check that the IP header length is greater than or equal to the minimal Internet header length (20 octets), and a packet's total length is greater than its header length [18].
2. **IP Checksum**. Detection of corrupted packets can be useful for packets that have other IP standard violations to be able to adjust the results properly.
3. **IP Address**. We look for the following cases of the IP address field:
   - Private IP addresses which ideally should not appear in public internets but do exist in the public domain (both from our experience and related literature [16], [4]);
   - Spoofed IP addresses that can be clearly identified in the global internet, e.g. so-called "land attack" [22], [9].

[1] TCP traffic comprises about 98% of packets on the global link and at least 60% on the local network on average.

- Certain special cases of IP addresses that can not be used as either source, destination, or either kind of address on a public internet [21].
- Packets in which both the source or destination IP addresses do not belong to the OU address space since all packets on the monitored links are expected to come to or from the university.

4. **IP Options**. The source routing option [18] typically should not appear in modern internets. Other IP options, at the time of this writing, are not known to have potential to harm the destination since implementations are supposed to discard unknown options if any are present in a packet [18]. Truncated options also should not appear in valid packets, as they indicate that the entire IP header is not present in one packet, thus making the packet invalid.
5. **Other**. We also considered packets with small Time-to-Live values and overlapping data for this work but they were not included in the final analysis due to various reasons. See [3] for details.

C.2 TCP Header Analysis

1. **Packet Size**. Unfragmented IP packets are required to be large enough to hold an entire TCP header. In the case of fragmentation, the required part of the TCP header (20 octets) is normally present entirely in one IP datagram. Splitting TCP headers might be used to pierce firewalls.
2. **TCP Checksum**. Invalid TCP checksums might be used in subtle attacks where an attacker is aware of the presence of a monitor between them and the victim machine and tries to convey their activity undetected (described in detail in Bro [16]). TCP checksum verification can also be useful for packets that already have other TCP violations to determine whether the packets should be taken into consideration.

Not all of our packet traces include entire packets and a number of packets from the traces are truncated. Truncated packets were excluded from the statistical analysis of corrupted packets.

3. **Port Numbers**. Virtually any combination of source and destination port numbers can be valid except the reserved number zero.
4. **TCP Flags**. Only a few combinations of the six TCP flags can be carried in a TCP packet. Since URG and PSH flags can be used only when a packet carries data [19] thus, for instance, a combination of SYN and PSH becomes invalid. Moreover, any combination of more than one of SYN, RST, and FIN flags is also invalid[2]. Illegitimate combinations of TCP flags are known to be used in so-called "Xmas Tree" scanning and operating system detection techniques [7].
5. **Reserved bits**. The original TCP specification reserves six bits in the TCP header for future use. More recent extensions to TCP [20], [11] utilize some of those bits, but they are mostly experimental documents at the time of this writing[3]. Setting the reserved bits to an arbitrary value might harm poor TCP implementations.

[2]According to the T/TCP RFC [2], a packet that includes both SYN and FIN flags might be valid if it carries a CC or CC.NEW option. In our analysis, we take into account these options even though the implementation of T/TCP is experimental and is not a current standard.

[3]Explicit Congestion Notification (ECN) [20] has since become a proposed standard, but was not so at the time of capturing our packet traces.

Type		Global Link		Local Link	
IP	Private addresses	13,830	22.2%	2,703	1.1%
	Outside of range addresses	283	0.5%	244,833	98.0%
	Other address violations	280	0.4%	0	0.0%
	Improper options	0	0.0%	0	0.0%
	Too short packets	0	0.0%	0	0.0%
	Same src and dst addresses	0	0.0%	0	0.0%
TCP	Invalid TCP flags	196	0.3%	51	0.0%
	Zero port number	136	0.2%	6	0.0%
	Non-zero reserved bits	1,047	1.7%	61	0.0%
	Too short packets	0	0.0%	0	0.0%
	Invalid cksums	46,466	74.6%	2,178	0.9%
	Total errors	62,287	100.0%	249,838	100.0%
	Total packets	247,87,366		54,696,049	
	Error rate	0.025%		0.457%	

TABLE I

ERRORS DETECTED ON GLOBAL AND LOCAL LINKS

D. Analyzed Data

During our experiments we analyzed traces gathered November 2000 through June 2001 at different times of the day on two links described above. Each trace file consisted of several million packets and the total number of analyzed packets exceeded 300,000,000. The total number of reported warnings over all of the analyzed data was approximately 300,000, with over 75% of them coming from one trace file.

III. RESULTS

All errors recorded on the global and local links are summarized in Table I. It can be seen that the system did not observe all known types of violations, which tells us either that the amount of data analyzed was not large enough to detect such packets and calculate their rate or that they do not exist in large numbers on the Internet. Table I also shows the total number of packets analyzed and the number of errors generated for both links for comparison. Note that the error rate on the global link reflects a more realistic number than the error rate on the local link because the latter number is greatly influenced by one type of errors.

The nature and content of the traffic from the two monitored links differ substantially, which directly affects the number and type of errors obtained from each link. We performed analysis of the packets from each link separately because of this. Levels of detail with which we present material differ due to space limitations. More thorough analysis is discussed in [3].

A. IP Analysis

A.1 Private IP Addresses

Global Link

Results obtained during our experiments at the global link showed a large number of packets sent either to or from private IP addresses (see Table I). They were either TCP, UDP, or ICMP packets and the great majority of them belonged to the following types (listed in decreasing order):

Physical Address	Network Address	Packets	Percent
Local to Local	Private to Broadcast	2,595	96.0%
Router to Local	Private to Local	69	2.6%
Local to Router	Local to Private	39	1.4%
Total		2,703	100.0%

TABLE II
DISTRIBUTION OF PACKETS CONTAINING PRIVATE IP ADDRESSES CAPTURED ON LOCAL NETWORK

No	Category	Packets	Percent	Cause
1	Packets from private Microsoft IPs	7,260	3.0%	Microsoft OS specifics
2	DHCP packets from aol.com IP address	8,490	3.5%	Erroneous software
3	Limited broadcasts from the router	228,054	93.1%	Distributed DoS attack
4	Other	1029	0.4%	Unknown
	Total	244,833	100.0%	

TABLE III
CATEGORIES OF PACKETS WITH ADDRESSES OUT OF THE OU ADDRESS RANGE OBTAINED ON LOCAL LINK

- Attempts to establish TCP connections (mostly locally generated) and other TCP packets with no payload;
- UDP NetBIOS name service packets targeting private IPs;
- UDP DHCP broadcasts from private addresses;
- ICMP Echo Requests sent to private addresses.

The logged packets contain IP addresses that belong to all classes of private networks — A, B, and C — with roughly similar numbers in each category. We found that packets destined for private IP addresses are sent by various OU hosts that run different operating systems and have different configurations. Thus the presence of such packets can not be wholly explained either by errors in implementation or by improper default configuration of a certain operating system. We also determined that the majority of the packets coming from private addresses were sent to private destination addresses as well, which makes the probability of address spoofing smaller.

Local Network

The distribution of packets containing private IP addresses that were captured on the local link is shown in Table II. Since private IP addresses are valid on the network of their origin, we do not examine the first category. Note, however, that these broadcast packets did not trigger a response and thus are not likely to belong to valid connections.

The majority of the remaining packets were TCP packets sent by four external web servers from private addresses to local hosts and responses to them. This looks like a problem in a specific implementation. Other packets captured on the link included UDP NetBIOS and ICMP packets (most likely caused by misconfiguration) and unsolicited TCP packets with no data (could have been the result of a software error and backscatter packets[4]).

A.2 IP Addresses Outside the OU Address Range

All of the packets on the global link are expected to come either to or from the OU address space, while packets on the local link should have at least one of the two IP addresses from that range. Thus, all other packets are worth examination.

Local Network

The packets from the local link for which neither the source nor destination IP address were in the OU address space comprise 98% of all errors from the link and are categorized in Table III.

- **Case 1:** This group includes packets sent by local hosts from IP addresses in the range 169.254.0.0/16. The range belongs to the Microsoft-reserved class B network and is used by Windows hosts during so-called "automatic configuration" [13] if they are configured to use DHCP [5] and cannot obtain an IP address from a DHCP server.
One type of packets sent by these computers with private Microsoft addresses is DHCP Inform messages. Such messages should not be issued on networks with dynamic address assignment by we, in turn, did not observe DHCP Discover messages sent by those hosts. We cannot prevent packets that come from the private Microsoft addresses from leaving the network and therefore might want to treat them as private and apply additional rules to router filters.
- **Case 2:** The next interesting group is composed of packets sent from 172.128.x.x – 172.186.x.x IP addresses which belong to aol.com. The great majority of the packets are DHCP Inform packets sent to the limited broadcast IP address. Since America On-line (AOL) client software uses IP addresses from the AOL address space even in cases when a client host has already been preconfigured with a globally valid IP address [1], we suspect that local machines that were unable to obtain their global IP addresses from the DHCP server tried to use other IP addresses, i.e. assigned to them by AOL software, to obtain other network parameters. While the cause of the packets issued from the above mentioned IP addresses could be specifics in implementation of the AOL client software, hosts should not sent DHCP Inform packets on networks where address assignment is performed dynamically.
- **Case 3:** The largest group of packets with IP addresses outside the OU range came to the monitored network from various external IP addresses and targeted the limited broadcast address. Our first traces included very few packets of this type but one trace included an enormous number of ICMP Echo Request packets that entered the link and triggered an even greater number of ICMP Echo Replies in response. The Echo Requests came from about 25 different IP addresses with domain names in at least 12 countries[5]. Even though our router allowed the limited broadcast packets to come through[6], forwarding of such packets should normally be disabled on routers. We believe that

[4]For more information about backscatter analysis see [14].

[5]We were unable to resolve the names of all IP addresses.

[6]After obtaining this trace, limited broadcast forwarding was disabled at the router and similar cases do not appear in the following traces.

Case	Description	Used As	Packets
1	Specified host on this network	Destination	153
2	Limited broadcast	Source	127
	Total number of packets		280

TABLE IV
DETECTED IP ADDRESS VIOLATIONS ON GLOBAL LINK

Category	Packets	Possible Cause
Malformed packets	33	Corruption
ACK packets to port 6	98	Possible attack
Other	5	Misconfiguration or other errors
Total	136	

TABLE V
TYPES OF PACKETS WITH ZERO PORTS ON GLOBAL LINK

these packets could not have originated outside of the university (they would have been discarded at our ISP), but rather came from an OU machine and had their IP addresses spoofed. Since the number of Echo Requests that we captured was very large and the number of responses was thousands of times larger, we, with some degree of certainty, can conclude that all machines on the local network participated in a distributed DoS (DDoS) attack against the above mentioned machines.

Global Link

Analysis of the global link showed a smaller number of packets of this type and fewer packet types than on the local monitoring point because not all of such packets propagate to the global link. The global link packets contained the private Microsoft and aol.com addresses all of which were addresses above. An interesting observation is that 85% of the packets that had both the source and destination addresses out of the OU address range were sent during morning hours on weekdays. This could be a bootstrapping issue, even though we were not able to verify the fact due to limited information available about the sending hosts.

A.3 Other IP Address Violations

The remaining IP address violations come from so-called “special” IP addresses that may not be legitimately used as either the source (broadcast), destination (“this network”), or possibly either kind of address (loopback) [21]. We look for five different types of such IP addresses (detailed in [3]) and present our results here. None of these types of violations were detected on the local link, and therefore we provide global link data only.

Global Link

Table IV lists global link results and shows that the system captured only two out of five error types.

Case 1 shows packets that were sent to network 0 and therefore could not be routed to the destination. The great majority of these packets are SYN packets sent to well known TCP port numbers (25, 80, and 524), while others were UDP NetBIOS packets. We believe that both types of erroneous packets were caused by misconfigured software, and a number of examples from the traces support this opinion. All of the packets combined in this category belong to outgoing traffic and are not known to be either dangerous or useful.

Case 2 provides statistics for packets sent from the IP address 255.255.255.255. The majority of these packets were ICMP UDP Port Unreachable messages sent to private Microsoft addresses (169.254.0.0/16) and reported different port numbers with the most common port being 2519. We believe that these packets were sent in response to UDP packets broadcast from the private Microsoft addresses on local networks by a computer or another device with a poorly implemented IP stack that used the limited broadcast address as the source address.

Other packets from this group were TCP RST packets sent from 255.255.255.255 where some of them inadvertently allowed us to detect a number of large network scans. During these scans, a SYN packet was sent to a particular port on every host on a network. Some SYN packets sent to network addresses 0 triggered replies back from the IP address 255.255.255.255 and, similarly to the UDP Port Unreachable messages, could have been sent by a poorly implemented device that responded to packets intended for network IP addresses.

B. TCP Analysis

B.1 TCP Packets with Zero Ports

Since the rate of packets with zero ports is low, the system logged only six such packets on the local network, mostly corrupted, and we omit their description here.

Global Link

Table V summarizes all of the categories of packets with zero port numbers captured on the global link. The first group represents packets that were either corrupted in the network or at the sender. A significant number of them were malformed in a similar way and likely caused by an error in TCP implementations. See [3] for details.

The next group of packets looked very strange, and all of them followed a very specific pattern. The packets were simple ACKs, had the source and destination ports 0 and 6 respectively, and advertised a TCP window of size 0. The majority of these packets were sent in groups logically increasing the sequence numbers they acknowledge with about 6% of them sent from private IP addresses. The number of packets with ports 0 and 6 was rather large and these suspicious packets could have been sent using the same tool or application.

B.2 Invalid TCP Flags

Global Link

Table VI lists all of the groups of packets with invalid combinations of TCP flags as seen on the global link. We do not perform any further analysis on the first two groups but packets from the third group were issued:

1. As RST packets in response to an initial SYN or other unsolicited packet;
2. After a FIN packet in the same direction where the other end did not close the connection after 3–5 minutes.
3. After a RST packet in the same direction.

Type	Packets	Percent
Failed TCP checksum	74	37.8%
Malformed packets from existing connections	108	55.1%
FIN RST to terminate a connection	12	6.1%
PSH set in SYN	1	0.5%
Probe packets	1	0.5%
Total	196	100.0%

TABLE VI
TYPES OF PACKETS WITH INVALID TCP FLAGS ON GLOBAL LINK

There is no single explanation for all of these packets. A number of them, especially responses to SYN and unsolicited packets, were most likely legitimate packets and meant by the sender to be pure RST packets.

The last packet from Table VI had the SYN, FIN, URG, and PSH flags set. It was issued along with other apparent probe packets which we believe were part of a fingerprinting attempt[7]. Thus, our empirical results suggest that a very small fraction of packets having invalid combinations of TCP flags are likely to be a result of malicious activity.

Local Network

The rate and types of packets having invalid TCP flags on the local link is conformant with the rate and types of similar packets from the global link, and we do not categorize them in this paper. An interesting fact is that the majority of the packets with failed checksums had the SYN, FIN, RST, and URG bits set, carried payload, and were sent from port 18245 to 21536 (such packets also appear on on the global link but are not as common) and looked like a specific implementation problem.

B.3 TCP Header Reserved Bits

According our analysis, only a very small portion of packets with TCP reserved bits violations can be viewed as sent intentionally and we omit discussion of such packets here due to space limitations. See [3] for more information.

IV. PACKET DISTRIBUTION ANALYSIS

This section summarizes results of our analysis and provides distributions of erroneous packets over the time of day and by the possible cause.

A. *Packet Distribution Over Time*

The packet traces that we included in this study were captured at different times of the day: the data gathering process always started at 3am, 6am, 9am, 3pm, and 11pm on the global link, but was not that systematic at the local link. This allows us to easily build error rate distribution over time for the global monitoring point (plotted in figure 1) but would be difficult to do for the local link. Figure 1 shows that the error rates are always higher during business and evening hours and drop significantly at night. Since a large number of erroneous packets were presumably caused by misconfiguration, we expect that a lot of software that issues such packets is unlikely to be system software that runs constantly. Also, a lot of errors on the link are due to failed checksums while errors in data transmission are more likely to occur at the day, when networks are busy. Therefore, we believe our results are realistic.

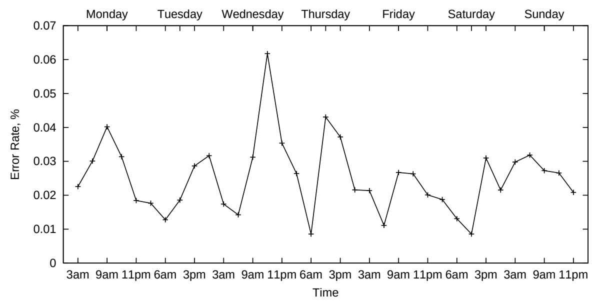

Fig. 1. Distribution of Erroneous Packets by Day of Week on Global Link

Type	Errors	Error Rate	Fraction of Errors	Fraction of Packets
Incoming	2,306	9.35×10^{-5}	10.6%	45.0%
Outgoing	101	3.89×10^{-6}	0.5%	47.6%
Local	19,377	4.78×10^{-3}	88.9%	7.4%
Total	21,784	3.99×10^{-4}	100.0%	100.0%

TABLE VII
DISTRIBUTION OF ERRONEOUS PACKETS BY DIRECTION ON LOCAL LINK (ADJUSTED)

In order to study the influence of the packet direction on the distribution of error rate over time, we divide all packets into incoming, outgoing, or local (local link only). Table VII shows the error rate of packets that were recorded as erroneous at the local link for each data direction with the DDoS attack described in section III-A.2 being excluded. Table VIII shows the same distribution for the global link, and figure 2 plots the corresponding error rates over time. The majority of errors on the local link relate to invalid IP addresses, while the cause of many global link errors, according to [23], could be a faulty piece of hardware or flawed software in the forwarding path.

B. *Packet Distribution by Possible Cause*

Based on our analysis provided in section III, Table IX draws distributions of different categories of packets according their possible cause. It does not include the DDoS packets described in section III-A.2. From all of the categories listed in the table, only packets included in the "Malicious User" and "Unknown" groups might purport malicious intent. The "Malicious User"

Type	Errors	Error Rate	Fraction of Errors	Fraction of Packets
Incoming	5,844	5.00×10^{-5}	9.4%	47.1%
Outgoing	56,312	4.30×10^{-4}	90.6%	52.9%
Total	62,156	2.51×10^{-4}	100.0%	100.0%

TABLE VIII
DISTRIBUTION OF ERRONEOUS PACKETS BY DIRECTION ON GLOBAL LINK

[7] A popular network scanner `nmap` [6] is known to send packets with exactly the same combination of TCP flags when trying to determine operating system type of a remote host.

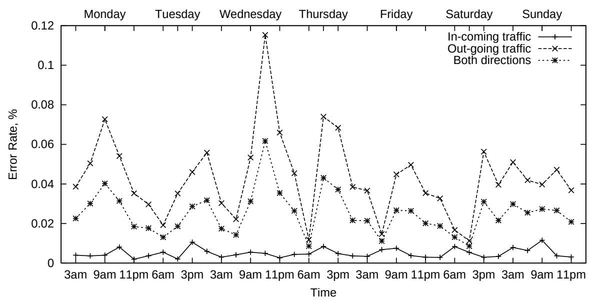

Fig. 2. Distribution of Erroneous Packets by Time for Different Packet Directions on Global Link

Type	Global Link		Local Link	
Legal Packets	451	0.7%	2,598	11.9%
Corrupted Packets	47,234	75.8%	2,286	10.5%
Poor Implementation	372	0.6%	15,840	72.7%
Misconfiguration	11,096	17.8%	16	0.1%
Backscatter Packets	1	0.0%	1	0.0%
Malicious User	18	0.0%	0	0.0%
Unknown	3,115	5.00%	1,043	4.8%
Total	62,287	100.0%	21,784	100.0%

TABLE IX

DISTRIBUTION OF ERRONEOUS PACKETS BY POSSIBLE CAUSE

category consists of only those packets that we, with some degree of certainty, view as sent intentionally (i.e. probe packets from various scans). The packets that could either be issued by an attacker or had a different origin (and we were unable to determine the cause of such packets) are united as "Unknown". All other categories, to the best of our knowledge, are either the result of a mistake or are not related to human factors, and therefore we believe that they were not sent by an attacker.

Table IX shows that only 5% of all packets that triggered warnings on both links might signal malicious intent. In reality, the actual number of packets sent by attackers is smaller than the number of packets listed in these two categories. According to our analysis, the largest portion of erroneous packets on the local link was caused by incorrect implementations, while the same category is under 1% of all errors on the global link. This tells us that such packets are not likely to propagate far beyond the link of their origin. And only a small portion of all packets recorded by the system can be considered intrusive.

V. CONCLUSIONS

In many cases, it would be beneficial to secure a site at the router against possible attacks that use invalid values of IP or TCP header fields. Suggested router filters have been intensively discussed in prior literature, and we omit their description here. Such filters block both some attack packets and packets malformed by a mistake which are neither harmful nor useful.

Taking into consideration proper router filters, one can see that there are not a large number of malformed packets on the network. A significant increase in the number of such packets might, however, indicate a problem — the DDoS described in this work had the same impact on the packet error rate. A dramatic increase in the number of erroneous packets might be used as an additional indication of an attack for an IDS that catches a DoS attack based on its attack signatures.

During our analysis we discovered a number of unexpected things, some of which are not well understood. They include:

- Unusual usage of the private Microsoft and aol.com addresses;
- Strange communication on ports 0 and 6;
- Packets sent from the address 255.255.255.255 in response to packets sent to network addresses.

We also found a number very specific implementation problems and could correlate an increase of errors during morning hours to a bootstrapping issues. In some cases, however, the amount of data we had was not sufficient to determine the cause of a problem, and having more (and more diverse) data might have helped this job.

VI. ACKNOWLEDGMENTS

We would like to thank Mark Allman and Ethan Blanton for useful comments and suggestions on earlier drafts of this paper and Terry Kelleher for providing us with data files. Thanks also go to anonymous reviewers for suggested improvements of this work.

REFERENCES

[1] America Online, Inc. *Webmaster Info, Article 14*, October 2001. http://webmaster.info.aol.com/index.cfm?sitenum=2&article=14.
[2] R. Braden. T/TCP — TCP extension for transactions functional specification. RFC 1644, July 1994.
[3] M. Bykova. Statistical analisys of malformed packets and their origings in the modern internet. Master's Thesis, March 2002. http://irg.cs.ohiou.edu/papers/.
[4] CAIDA. Internet measurement: Myths about internet data, July 2002. http://www.caida.org/outreach/presentations/Myths2002.
[5] R. Droms. Dynamic host configuration protocol. RFC 2131, March 1997.
[6] Fyodor. `nmap`. http://www.insecure.org/nmap/.
[7] Fyodor. Remote OS detection via TCP/IP stack fingerprinting, October 1998. http://www.insecure.org/nmap/nmap-fingerprinting-article.html.
[8] M. Handley, V. Paxson, and C. Kreibich. Network intrusion detection: Evasion, traffic normalization, and end-to-end protocol semantics. In *10th USENIX Security Symposium*, August 2001.
[9] Internet Security Systems. *Real Secure Attack Signatures*, 2000.
[10] V. Jacobson, C. Leres, and S. DMcCanne. `tcpdump`, June 1989. http://www.tcpdump.org.
[11] R. Ludwig and R. Katz. The Eifel Algorithm: Making TCP robust against spurious retransmissions. *ACM Computer Communications Review*, 30(1), January 2000.
[12] M. Mahoney and P. Chan. PHAD: Packet header anomaly detection for identifying hostile network traffic. Technical report, Florida Tech., technical report CS-2001-4, April 2001.
[13] Microsoft Corporation. *Automatic Private IP Addressing*. http://www.microsoft.com/TRAININGANDSERVICES/content/training/samples/2152/Webfiles/Mod03/03m10.htm.
[14] D. Moore, G. Voelker, and S. Savage. Inferring internet denial-of-service activity. In *10th USENIX Security Symposium*, August 2001.
[15] S. Ostermann. `tcptrace`, 1994. http://www.tcptrace.org.
[16] V. Paxson. Bro: A system for detection network intruders in real-time. *Computer Networks*, 31(23–24):2435–2463, December 1999.
[17] V. Paxson, M. Allman, S. Dawson, W. Fenner, J. Griner, J. Heavens, K. Lahey, J. Semke, and B. Volz. Known TCP implementation problems. RFC 2525, March 1999. http://www.ietf.org/rfc/rfc2525.txt.
[18] J. Postel. Internet protocol. RFC 791, September 1981.
[19] J. Postel. Transmission control protocol. RFC 793, September 1981.
[20] K. Ramakrishnan, S. Floyd, and D. Black. The addition of explicit congestion notification (ECN) to IP. RFC 3168, September 2001.
[21] J. Reynolds and J. Postel. Assigned numbers. RFC 1700, October 1994.
[22] R. Sekar, Y. Guang, S. Verma, and T. Shanbhag. A high-performance network intrusion detection system. In *Proceedings of the 6th ACM conference on Computer and communications security*, pages 8–17, 1999.
[23] J. Stone and C. Partridge. When the CRC and TCP checksums disagree. In *ACM SIGCOMM 2000*, August 2000.