Examining Intrusion Prevention System Events from Worldwide Networks

Sathya Chandran Sundaramurthy*
HP Labs
Princeton, NJ, USA
sathya@ksu.edu

Sandeep Bhatt
HP Labs
Princeton, NJ, USA
sandeep.bhatt@hp.com

Marc R. Eisenbarth
HP TippingPoint
Austin, TX, USA
marc.r.eisenbarth@hp.com

1. ABSTRACT

We report preliminary results on analyzing a large dataset of over 35 billion alerts recorded over a 5 year period by Hewlett-Packard (HP) TippingPoint Intrusion Prevention System (IPS) devices located in over 1,000 customer networks worldwide. This dataset provides a rich view into the nature of attacks, both external and internal, across diverse networks. This paper presents our initial findings. For example, (i) while most customers are among the early victims of only a handful of attacks, a few customers are early victims of a large number of attacks, (ii) vendor vulnerability disclosures sometimes lead to a surge in exploit attempts, and (iii) even after a decade, some worms such as Slammer show very significant spikes in their activity and infection rates.

Categories and Subject Descriptors

A.1 [**General Literature**]: Introduction and Survey; C.2.0 [**Computer-Communication Networks**]: Security and protection

General Terms

Documentation, Measurement, Security

Keywords

Big data analysis, HP TippingPoint, Intrusion Prevention System, Threat analysis

2. INTRODUCTION

HP TippingPoint IPS devices are installed in over 1,000 customer networks, deployed in every part of the network, from the perimeter to the network core, and see a very wide variety of attacks, from common everyday attacks such as Cross Site Scripting (XSS) to the more sophisticated attacks against Microsoft RPC bugs being launched within a network by a malicious insider. The IPS devices inspect traffic in real-time and enforce security policies at network speeds approaching 10 Gbps with over 6,000 filters deployed. The policies are maintained as part of the TippingPoint Reputation Digital Vaccine (DV) service. Figure 1 shows a typical deployment of a TippingPoint IPS in a customer's network.

Figure 1: TippingPoint IPS deployment

IPS devices use signatures to flag inbound and outbound traffic that is known to be bad or that violates policy. As new vulnerabilities become known, new signatures are pushed out to IPS devices. When an incoming or outbound packet triggers a filter, the filter can either block, rate-limit or allow the traffic to pass through. TippingPoint devices are configured to block all traffic that triggers critical and high severity alerts. In all cases, an alert is recorded by the device; among other fields, each alert contains the source and destination IP addresses, port numbers, filter ID, a hit-count which represents the number of times the filter was triggered within a one-minute interval, and timestamp.

In contrast to Intrusion Detection Systems (IDS), IPS filters which block traffic must be guaranteed to have very low false positive rates. This generally comes at the expense of potentially higher false negative rates. Low false positive rates are achieved by designing filters to block post-compromise events such as a connection back to a command and control server or traffic intended to propagate the infection to other machines. Furthermore, the filter set focuses on blocking vulnerabilities rather than specific incarnations, often called exploits. This difference is important as well in

*Sathya Chandran is a Ph.D., student in the Computing and Information Sciences department at Kansas State University, Manhattan, KS, USA. This work was done as part of his summer (2012) internship at HP Labs, Princeton, NJ, USA.

BADGERS'12, October 15, 2012, Raleigh, North Carolina, USA.

that a generic signature for, say, the MS08-067 vulnerability will block any exploit variant that produces the network traffic on the wire needed to exploit this vulnerability, as we see for example with the numerous Conficker variants. In other words, the IPS serves to block the behavior common to all exploits that leverage a given vulnerability, not necessarily the exploits themselves.

Clearly, no single mitigation mechanism can address the myriad of complex attack scenarios that challenge computer security today. Often times, NIPS must pass on complex malware analysis, thereby allowing an endpoint to be infected and instead focus on detecting and preventing any resulting outgoing communication for this host to command and control servers. With the complexities associated with compound documents such as PDF and Microsoft Office documents, and the plethora of vulnerabilities against these code bases, the most reliable defense is to identify machines post-compromise for remediation and excise traffic associated with compromise emanating from these machines.

TippingPoint customers can opt-in to allow the aggregated alert data to be sent to a centralized ThreatLinQ server which provides aggregate statistics and common lists of bad IP addresses and domain names. This ThreatLinQ dataset we have started to analyze has been collected over a 5 year period, between 2007 and early 2012. Besides its large size, processing the data required non-trivial effort because of the complexity of interpreting, analyzing and parsing the accompanying filter metadata and correlating it with the raw alert data.

Our goal in examining this dataset, beyond reporting aggregate statistics, is to understand the nature of attackers, the attacks launched, and the customers targeted. For example, are certain customers, or groups of customers, more likely to be early targets of many attacks? Are there relationships between attacks and their targets? Do attacks occur in clusters? This extended abstract reports our initial findings. As we continue to mine the dataset we expect to develop furher insights into the nature of attacks.

3. DATASET CHARACTERISTICS

The dataset consists of alerts from TippingPoint IPS devices deployed in roughly 1,000 customer networks worldwide, between 2007 and 2012. The IPS device detects malicious traffic through the use of filters that monitor the network flow and take predefined actions when a malicious flow is detected. In all, we observed 3,834 unique filters that triggered roughly 35 billion alerts at the IPS devices.

An alert contains the following eight fields: filter ID, attacker and victim IP and port numbers, hit count (number of times this filter fired for this flow in the past 1 minute), customer ID and timestamp. Each filter has a unique ID, and is assigned a severity rating of low, medium, high or critical. In addition to the alert data, we also obtained comprehensive information about each of the filters explaining in detail the attacks they detect, the severity of the attack detected, references to vendor disclosure of vulnerabilities, and this information was utilized during different case studies we did on the dataset.

Some of the difficulties in making strong conclusions from this dataset included (i) customers were online at different times, (ii) IPS devices placed in different parts of the network for different customers (maybe they were moved as well), and (iii) different customers had different blocking capabilities upstream of the IPS devices.

3.1 Filter and alert classification by severity

Severity denotes the criticalilty of the attack detected by the filter. For example, a filter that detects shell code in the payload portion of a packet sent to a web server will have higher severity than the filter that detects a port scan for a particular port. By default, when a critical filter is triggered, the corresponding flow is blocked. Figure 2 gives the fraction of each of the filter types seen in our dataset and the fraction of total alerts contributed by filters from each category. Some of the filters in the dataset were no longer being used by TippingPoint and are thus deprecated.

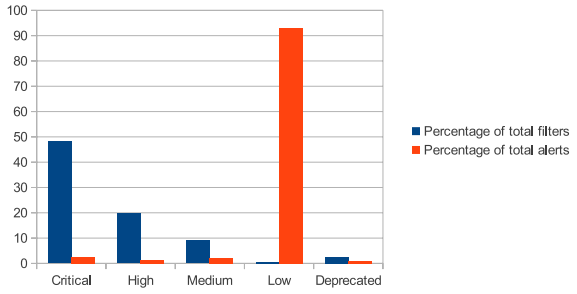

Figure 2: Statistics for filters by severity

3.2 Attack trend over the five years

Figure 3 shows the number of alerts triggered for critical and high severity filters over the five years over all customers. We excluded the filters that detect Slammer attack in this plot as we analyze those in a separate section. Of the top ten peaks that can be seen in 2011, we wanted to find out the filters that generated the maximum alerts for each peak. Of the 10 peaks, the filter to detect a LAND attack, a well known denial-of-service attack first seen in 1997, where a packet's source and destination addresses are set to the victim's IP address causing the machine to reply to itself continuously, constituted the maximum alerts for 8 of them. Filters to detect packets where the destination address is set to loopback address and Back Orifice communications accounted for the other two peaks.

3.3 Analysis of attacker IP addresses

There were in all 9,403,495 unique attacker IP addresses of which 96.51% are routable and the remaining 3.49% are non-routable. We did an analysis to find out the relationship between each attacker IP address and the number of customers targeted. We do not include the non-routable addresses since these are, in general, common across multiple customers and including them would lead to erroneous results. Of the total attacker IP addresses, 3.49% targeted one customer and the remaining 96.51% launched attacks against more than one customer. A detailed statistics of this analysis is shown in Table 1. The analysis shows that there was no IP address that launched an attack on more than 300 customers and the majority of them targeted between 2 and 5 customers.

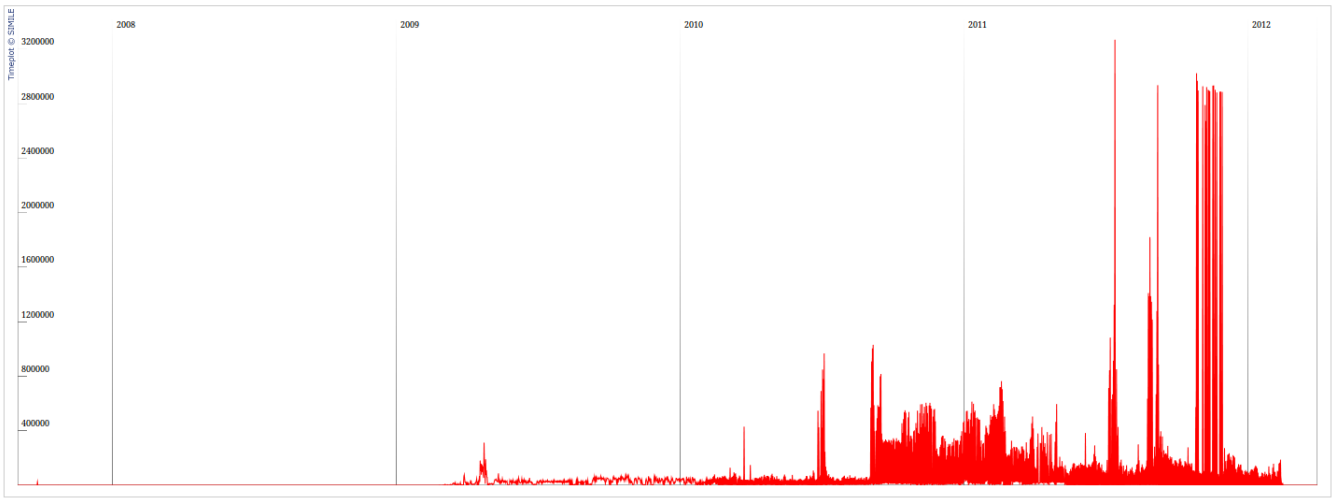

Figure 3: Distribution of critical alerts over five years

# of customers	# of IP addresses
2-50	9,073,918
51-100	1,498
101-150	137
151-200	19
201-250	6
251-300	3

Table 1: Relationship between attacker IP address and number of targeted customers

Of the 9 IP addresses that attacked more than 200 customers, 6 of them belonged to an ISP in China, 1 each to ISPs in Indonesia, the USA and Poland. Also 4 of these IP addresses were blacklisted in atleast two well known public blacklists. It is important to note that the default deployment of the filters seen in these attack campaigns result in blocking the offending Layer 7 attacks, not the more coarse-grained control of blacklisting the offending IP address.

3.4 First hit time analysis

For the high and critical severity filters, we asked the following questions.

- For each filter, how many customers were simultaneously targeted when the filter first fired?
- For each customer, how many times was it among the first to be targeted by an attack (filter)?

In order to make this calculation manageable, we further aggregated the data into windows of 12 hours. For each customer-filter pair we divide the entire five year period into buckets of 12 hours and aggregated the number of times every filter fired for each customer within each bucket. For a given filter, all customers who raise the corresponding alert in the first 12 hour window that the filter first triggered in are termed first-hit customers for that filter.

The results are shown in Figures 4 and 5. In Figure 4 we see that less than a dozen filters were triggered for more than 10 customers. In figure 5 we see that only three customers were early targets of over 100 attacks, while the majority of

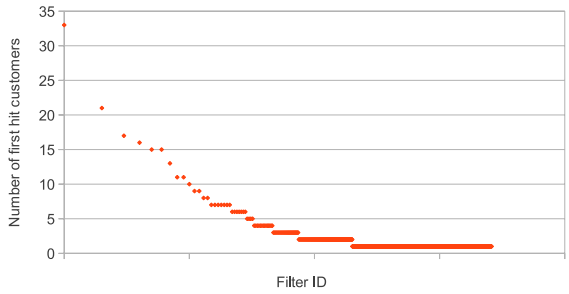

Figure 4: Distribution of number of first hit customers for each filter (critical and high severity). Filter IDs are plotted on a log-scale along the X-axis and the number of first-hit customers along the Y-axis.

customers were early targets for less than 10 attacks. This analysis identifies customers who were extremely vulnerable and may also help justify efforts to share attack information in real time.

The first analysis helped identify those attacks that were spreading globally as shown in Table 2. The second analysis will enable us to identify customers who were extremely vulnerable and may also quantify the advantage of sharing attack information across multiple customers.

4. CASE STUDIES

This section presents statistics on botnet and worm infections, including the notorious Slammer worm, various DDoS attacks, and the outbreak of attacks surrounding the release of vendor patches for some notable vulnerabilities.

4.1 Botnets

Botnets are used for spamming, stealing private information, and launching denial of service attacks. Different types of botnets emerged over the past five years and the IPS has filters to detect each of those specific bots in customers' networks. These are high-confidence filters that detect at-

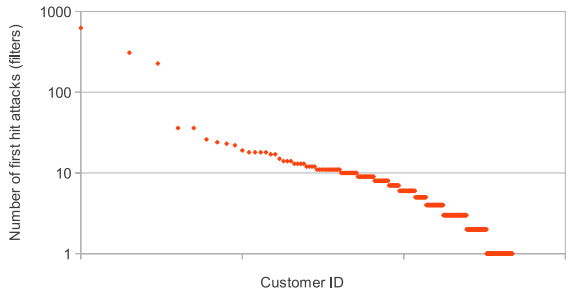

Figure 5: Distribution of number of filters (critical and high severity) for each of the first hit customers. Customer IDs and number of filters are plotted along the X- and Y-axes respectively, both on a log-scale.

Application Exploited	# of Customers targeted
Microsoft Office	33
Beagle mass mailing virus	21
Microsoft Excel	17
Microsoft Abstract Syntax Notation parser library	16
3COM Intelligent Management Center TFTP Server	15
Microsoft Internet Explorer	15

Table 2: Attacks that targeted 15 or more customers globally

tempts by infected machines to contact the command and control server (C&C server) of the attacker. Table 3 lists the fraction of infected customers and the number of alerts raised by five well-known botnets.

4.2 Internet worms

Table 4 lists some of the common worms, the fraction of infected customers and number of alerts raised for each worm. Many of these worms have been around for over 10 years. These filters are carefully designed to have zero false positive detection rates by detection behaviors specific to each of the worms.

The most striking entry is the Slammer filter, which raised more than a hundred times as many alerts as any other filter. In fact, Slammer accounts for almost 2% of all alerts raised by 6,000 filters over the 5 year period. The Slammer worm exploits a buffer overflow vulnerability in the Microsoft SQL Server 2000 resolution service running on UDP

Botnet type	Infected customers	Total alerts
Monkif	10.80%	134,685
Zeus	10.58%	603,903
Spy Eye	3.38%	40,868
Torpig	0.87%	1,274
Rustock	0.1%	7

Table 3: Botnet infection

Worm type	Infected customers	Total alerts
Slammer	52.29%	651,493,290
Nimda	46.28%	1,234,753
Back Orifice	31.44%	5,907,364
Storm	8.29%	24,204
Code Red	2.29%	48,047
SQL injection	1.31%	2,027,000
Code Red II	0.98%	17,404
Anig	0.32%	464
Phat bot	0.32%	423
Voyager	0.21%	348

Table 4: Worm infections

Attacker location	# of attacking IP addresses
United States	257,848
China	130,543
Canada	17,555
India	14,583
Russian Federation	12,770
Brazil	11,408

Table 5: Slammer Worm attacker location

port 1434 [1]. An infected host scans random IP addresses very rapidly in search of potential victims, sometimes causing significant network degradation in the process. Slammer was first noticed on January 25, 2003. We saw the first alert for Slammer in our dataset on January 23, 2009 and the last alert on February 14, 2012. As seen in Figure 6, the alerts peaked to a maximum of almost 42 million on February 15, 2011. There have been reports [2] that Slammer activity, which always exists in the background, dipped significantly between March 1 and April 12, 2011. This is consistent with our findings; it is likely that, in response to the February 15 spike, administrators initially took measures to weed out Slammer infections.

Many people have noted that Slammer persists on the Internet as a sort of background radiation and our results are consistent with this, except for a specific high volume denial-of-service attack using the Slammer payload targeting just one customer. While it is certainly possible that the target was a vulnerable instance of Microsoft SQL Server, it is also quite possible that the intended victim was a piece of security or networking equipment in hopes that it could not keep up with the attack volume.

The top origin countries of the attackers and victims and the number of unique IP addresses for each appear in Tables 5 and 6 respectively.

Victim location	# of targeted IP addresses
United States	1,596,374
Taiwan	391,550
Panama	277,679
Thailand	128,597
Netherlands	55,861

Table 6: Slammer Worm victim location

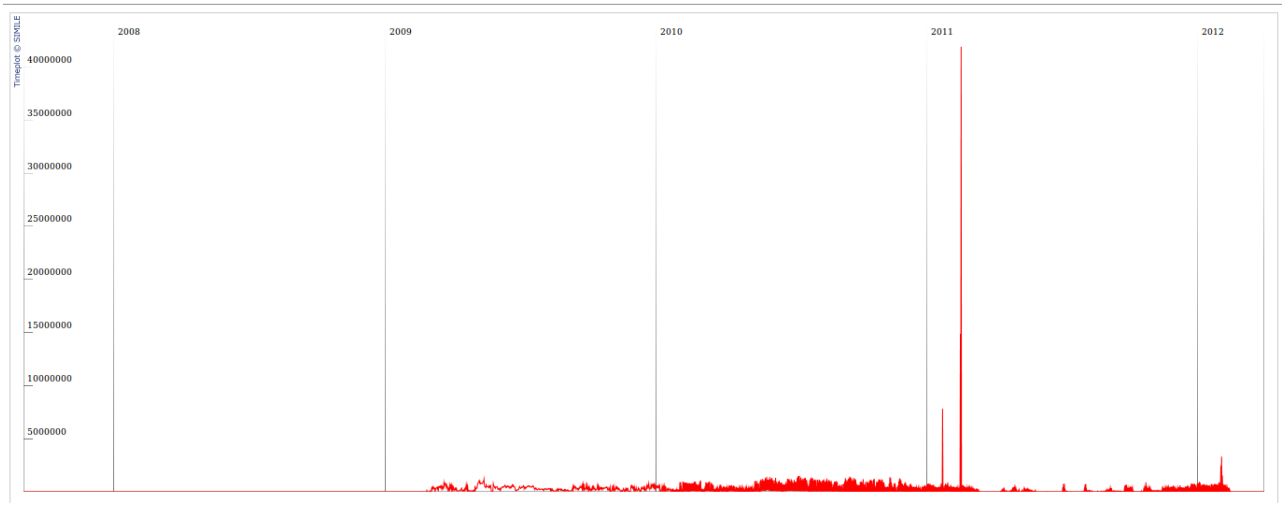

Figure 6: Slammer alert distribution

We also classified the IP addresses that were involved in the Slammer alerts. Of the roughly 600,000 publicly routable IP addresses that were found launching the Slammer worm, we found that 1,932 IP addresses were involved in only three types of alerts. The filters corresponding to those three type of alerts were (i) the Microsoft SQL Slammer-Sapphire Worm, and (ii) two filters to detect buffer overflow attack against Microsoft SQL Server Resolution Service. A reasonable conclusion is that these machines were customized as launching points for a wide scale Slammer attack.

4.3 Effect of patch release from vendors

As part of the Zero Day Initiative program, TippingPoint has access to a very large number of zero-day exploits. During the coordinated disclosure process, filters are written to protect customers against these vulnerabilities even before a patch is released by the vendor. Thus the filters associated with these vulnerabilities provide a unique insight into the true lifecycle of a vulnerability. Our data tracks the original discovery of the vulnerability, when it was disclosed to the vendor, when TippingPoint introduced a filter for this vulnerability, when the patch was released by the vendor and the infection lifecycle from this point onward in the form of filter telemetry data. This unique vantage point, combined with other notable external factors, such as the increased prevalence of analysis of vendor patches by attackers, which accelerates the infection, serves as the basis for the observations below. From a data analysis perspective, this requires correlating data from multiple sources. Here we have correlated the alert data with the filter metadata and the data from the Zero Day Initiative.

4.3.1 *Mozilla Vulnerability*

Mozilla announced on March, 2010 many JavaScript vulnerabilities such as [3, 4, 5] in their multiple products; Firefox, Thunderbird, and SeaMonkey. Their suggestion was to turn off JavaScript in their applications until the patch was released. Eventually, they released a patch on April, 2010. The IPS had a filter to detect an exploit against this vulnerability deployed on November, 2005. We wanted to see the alert distribution for this filter prior to and after the release of the patch. The result is shown in Figure 7. From this plot it is clear that the number of alerts increased significantly after the patch release date.

4.3.2 *Microsoft EOT Font Vulnerability*

The IPS has a filter that detects downloads of Extended OpenType (EOT) fonts. The EOT font is a proprietary font format from Microsoft that allows the embedding of fonts into web pages, along with digital rights management of the font information. There were a number of vulnerabilities [6] that could be exploited using EOT fonts including remote code execution. Microsoft released a patch on October 12, 2010, but the IPS had a filter deployed to detect the download of EOT fonts on January 10, 2006. As seen in Figure 8 the filter activity is moderate before October, 2010. But soon after the patch was released, there was a tremendous increase in the number of alerts; we believe that attackers became aware of this vulnerability and started hosting malicious websites that contain EOT fonts crafted and embedded in a way that would compromise Windows client machines. Even though the filter just detects the download of EOT font over the network (which could be benign), the fact that the download increased after a patch disclosure is suspicious.

4.3.3 *Symantec Client Security Buffer Overflow Vulnerability*

This filter detects an attempt to exploit a buffer overflow vulnerability found in certain versions of Symantec Client Security. Symantec Client Security contained a memory corruption flaw [7] in the Alert Originator service (iao.exe). The process blindly copies user-supplied data to a stack buffer via a memcpy call. By supplying a specially crafted packet, an attacker can overflow the buffer leading to arbitrary code execution in the context of the SYSTEM user. Symantec released a patch for this bug on January 27, 2011 but the IPS had a filter released on May 4, 2009. Even though the number of alerts seen for this attack is small (around 1,250 in all), it is worth noting that our data shows that all the alerts for this vulnerability occur after the patch release.

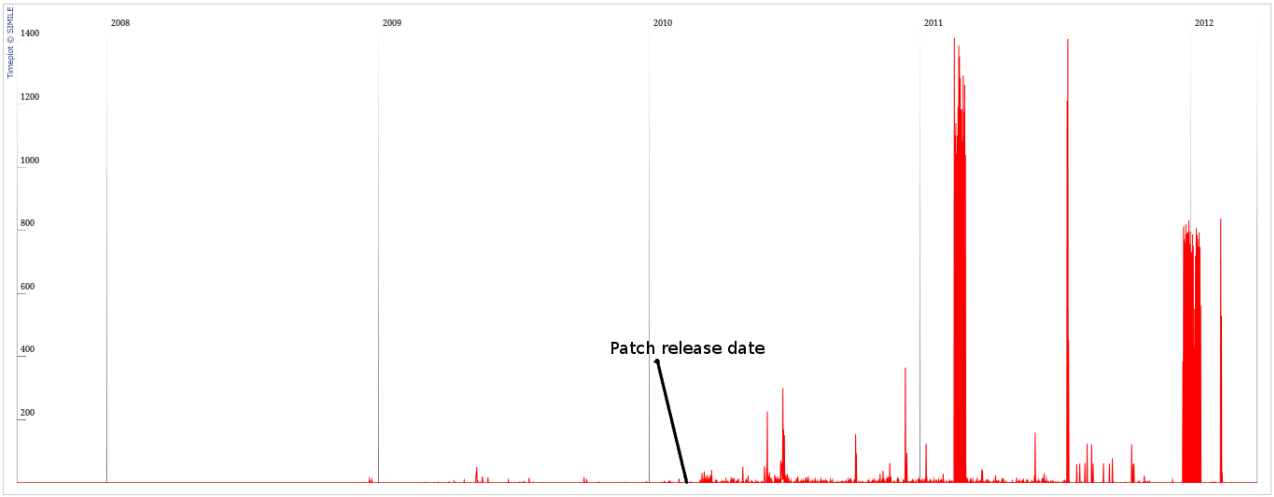

Figure 7: Mozilla vulnerability:Filter release date-November, 2005. Patch release date-April, 2010. The number of alerts increased after the patch release date, while there was very little activity for the prior years.

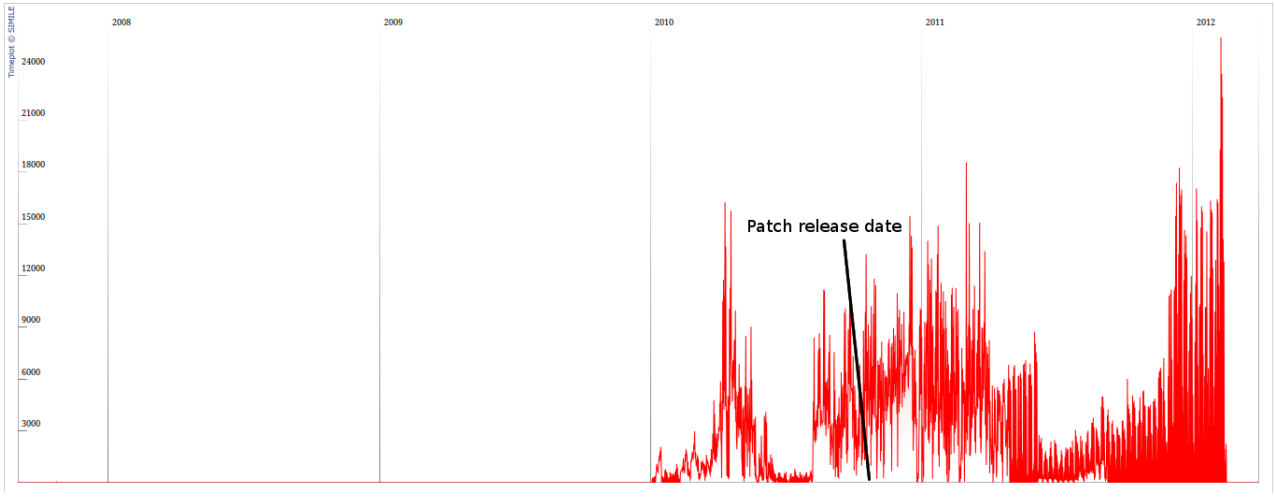

Figure 8: Microsoft EOT font vulnerability:Filter release date-January, 2006. Patch release date-October, 2010. Note the heightened activity after the patch release date.

4.4 Denial of Service

4.4.1 Layer 3 vs 7 DoS attacks

Total aggregated alerts for DoS and DDoS in the dataset is 6,876,051. The popular belief is that a denial of service is caused by sending a large number of packets in a small amount of time to choke the router or a target machine so that it becomes unavailable. But the recent trend is to use application vulnerabilities wherein a specially crafted packet is sent to the vulnerable application so that it must spend significant resources in order to process the request. We wanted to find out the fraction of total DoS alerts occupied by layer 3 and layer 7 DoS mechanisms. Layer 3 DoS alerts constituted 21.35% of total DoS alerts whereas the remaining 78.65% alerts where due to Layer 7 DoS attacks.

Appendix A lists the applications that were targeted for launching DoS attacks.

4.4.2 Low Orbit Ion Cannon (LOIC)

The notorious series of DOS attacks launched by the Anonymous group used a simple stress testing tool called LOIC. LOIC was mainly meant to stress test an application server and it does so by generating HTTP, TCP and UDP packets at a configurable packet rate. It is clear that UDP based DoS should be more effective than a TCP or HTTP based attack for the following reasons:

1. TCP-based attack must first establish a connection with the target through a three way handshake and then send the burst of packets. This slows down the effect of the attack.

2. Even if a TCP-based attack does not go through a three way handshake but sends a burst of TCP packets, a stateful firewall at the edge router can block the packets corresponding to unestablished flows.

3. A similar argument holds for HTTP-based attacks from

LOIC type	Fraction of total LOIC alerts
TCP based	99.96%
UDP based	0.04%

Table 7: Alerts for different LOIC versions

Attacker location	Alerts generated
Brazil	873,076
Colombia	13,779
United States	3,945
Anonymous Proxy	1,888
Switzerland	1,212
Spain	624
Canada	252
Venezuela	133
Germany	110
Sweden	57

Table 8: Top 10 LOIC attack attackers geographical distribution

LOIC, as HTTP runs over TCP predominantly (although there are variations where HTTP runs over UDP).

However, we found that the TCP variant was the most used attack vector. Table 7 presents the percentage of total LOIC alerts generated by each of the variants. We believe this discrepancy is due to the fact that among the options for flooding the network, TCP was the first option in the drop-down menu in the tool. So people who were using or tricked into using the tool didn't bother to change the default setting.

The maximum number of alerts for TCP-based LOIC attacks was found on January 22, 2012. Tables 8 and 9 give the attacker and victim IP addresses, respectively, mapped to geographical locations. We used MaxMind GeoLite [8] Country IP geolocation database to map IP addresses to countries.

4.4.3 Tribal Flood Network

There was one customer in the dataset whose machines had been used to launch a Distributed Denial of Service (DDoS) attack using the Tribal Flood Network (TFN) toolkit written by Mixter in 1999. Before we explain the events we provide a short overview of how the TFN operates [9].

The TFN is composed of three components: (i) attacker, (ii)client, and (iii)daemons. The attacker controls one or more clients and each client controls one or more daemons. The daemons are programs that launch packet based DoS attack against one or more victims as instructed by a client. An attacker communicates over TCP, UDP or ICMP with a client using either a Telnet or SSH shell piggybacking over these three network protocols. The client and the daemon communicate only through ICMP_ECHOREPLY packets. To run the client program on a client machine the attacker has to supply a list of IP addresses (daemons) that will be used to launch the DDoS attack and the type of attack. The supported types include SYN flood, UDP flood ICMP flood and Smurf attack.

Victim location	# of Alerts
Brazil	880,112
Colombia	10,976
United States	1,178
Spain	466
Canada	234
Netherlands	40
Sweden	20

Table 9: All LOIC attack victims geographical distribution

Coming back to our dataset, we noticed a number of buffer overflow attacks on multiple products installed in the customer's network machines. After the successful compromise of one or more machines the attacker should have installed the client and daemon programs. We saw the following list of alerts for communication between TFN daemons and clients.

1. TFN daemon acknowledging a *bind shell* command from a client. If the filter sees this response then the client has spawned a root shell bound to the TCP port specified by the client in the request.

2. TFN daemon acknowledging a *UDP Flood*, *TCP SYN Flood* and an *ICMP ECHO Flood* command from a client. When the filter sees this response then the daemon has started flooding the specified target using UDP, TCP SYN or ICMP ECHO packets.

3. TFN daemon acknowledging a *change packet size* command from a client. The TFN daemon under this request changes the size of the packets that it uses to flood the victims.

4. TFN daemon acknowledging a *Status/Stop* command from a client. The TFN daemon under this request reports its status and stops the flooding activity.

5. RELATED WORK

We are unaware of prior work on analyzing large Network Intrusion Prevention System (NIPS) datasets. While research on Network Intrusion Detection Systems (NIDS) has a rich history, we are also unaware of published studies that analyze NIDS alerts from a large number of customer networks over multiple years. Song et al. [12] developed a single-site traffic collection system to analyze NIDS data to characterize distribution of IDS alerts, origin of attacks and distribution of Anti Virus (AV) system alerts. Among their goals was to develop a large dataset of network traffic that researchers can use to analyze performance of IDS systems.

Recent research complementary to our paper, but related to the development of filters used within the TippingPoint IPS, is SandNet [11] reported in BADGERS 2011. This paper analyzes the long-term network behavior of over 100,000 known malware and characterizes their usage of prevalent protocols. This deeper understanding of malware behavior is aimed at developing better IPS filters.

Finally, Verizon issues an annual Data Breach Investigations Report [10] that analyzes data breaches across different industry groups. The reports are not based on network data alone, but on detailed incident reports that are filed by contributing businesses and state officials including the

US Secret Service, the Dutch National Tech Crime Unit, the Australian Federal Police, the Irish Reporting and Information Security Service and the e-Crime unit of the London metropolitan police. The annual report breaks down incidents by features such as industry, type of attack, and impact of the attack.

6. FUTURE WORK

While our preliminary findings are interesting in themselves, much more information can be mined from the dataset. We continue to look for significant correlations between attacks, and between attackers and groups of customers. We are also using the dataset to better understand vulnerability lifecycle.

We believe that analysis of this dataset can be useful in quantifying the benefits of collaborative security, an emerging research topic in intrusion analysis. The goal is to provide better early-warning services that can alert customers as attacks evolve over time.

7. ACKNOWLEDGMENT

We acknowledge Stuart Haber, Bill Horne, Pratyusa Manadhata and Prasad Rao of HP Labs, Princeton, for their valuable comments and suggestions throughout this work. We also thank Jason Jones of HP TippingPoint for providing us with the data we used for this work.

8. REFERENCES

[1] http://www.cert.org/advisories/CA-2002-22.html.

[2] http://blogs.mcafee.com/mcafee-labs/sql-slammer-worm-regains-momentum.

[3] http://www.mozilla.org/security/announce/2010/mfsa2010-17.html.

[4] http://www.mozilla.org/security/announce/2010/mfsa2010-18.html.

[5] http://www.mozilla.org/security/announce/2010/mfsa2010-19.html.

[6] http://www.microsoft.com/technet/security/Bulletin/MS09-029.mspx.

[7] http://www.symantec.com/business/security_response/securityupdates/detail.jsp?fid=security_advisory&pvid=security_advisory&suid=20090428_02.

[8] http://www.maxmind.com/app/geolite.

[9] http://staff.washington.edu/dittrich/misc/tfn.analysis.

[10] http://www.verizonbusiness.com/resources/reports/rp_data-breach-investigations-report-2012_en_xg.pdf.

[11] C. Rossow, C. J. Dietrich, H. Bos, L. Cavallaro, M. van Steen, F. C. Freiling, and N. Pohlmann. Sandnet: network traffic analysis of malicious software. In *Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security*, BADGERS '11, 2011.

[12] J. Song, H. Takakura, Y. Okabe, M. Eto, D. Inoue, and K. Nakao. Statistical analysis of honeypot data and building of kyoto 2006+ dataset for nids evaluation. In *Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security*, BADGERS '11, 2011.

APPENDIX

A. APPLICATIONS TARGETED FOR DENIAL OF SERVICE

Malformed JPEG image
SMB NetBIOS
Cisco IOS SNMP
IIS webDAV
IA Webmail Server
EMC Legato Networker
Computer Associates ARCserve Backup Message Engine
Windows 7/Server 2008 NetBIOS
Apple Webkit WebCore
Symantec Norton Antivirus
Quake 3
Windows RDP
Computer Associates BrightStor
OpenLDAP
BolinTech Dream FTP Server
MySQL XML XPath
Samba Flags2
Novell iManager
ISC DHCP Server
IBM Tivoli Storage Manager
Microsoft IIS Server
Microsoft Internet Explorer
Squid Proxy
OpenSSL
Sun JRE
Windows XP (IIS)
Microsoft Malware Protection Engine
SW-HTTPD Web Server
Microsoft SharePoint
Windows SMBv2
Microsoft Windows NETAPI