Surveillance Detection in High Bandwidth Environments

Seth Robertson[1], Eric V. Siegel[1], Matt Miller[1], and Salvatore J. Stolfo[2]

[1]*System Detection, Inc.*
{seth,evs,matt}@sysd.com

[2]*Columbia University*
sal@cs.columbia.edu

Abstract

In this paper, we describe System Detection's surveillance detection techniques for enclave environments (ESD) and peering center environments (PSD) and evaluate each technique over data gathered from two different network environments. ESD is evaluated over 74 hours of tcpdump *packet traces (344 million packets) from a large enclave; PSD is evaluated over 5 hours of* tcpdump *packet traces (110 million packets) gathered from a peering center. Both surveillance detection modules were executed over the audit data offline to generate surveillance detection alerts, though the systems can be run in real-time as well. Our results show that both ESD and PSD accurately discover great quantities of surveillance activities (including long-lived and distributed scans) and can be tuned to reduce the volume of alerts. Furthermore, existing IDS technology may be blind to many activities discovered by ESD and PSD.*

1. Introduction

Security software must detect surveillance activities to compete with the escalating sophistication and sheer prevalence of today's on-line attack procedures. Surveillance, the scanning of target IPs and ports for vulnerabilities, is the fundamental means to gather online attack intelligence, and is an increasingly common part of precise attack targeting. This is reflected by an alarmingly high proportion of connection attempts that are indeed surveillance probes, as measured and presented in this paper. The origins of such attempts range across most countries of the world, initiated by human attack activities as well as worms and otherwise captured drones. The range of technical strategies to perform surveillance is growing in variety and sophistication as methods become more precise and more stealthy (i.e., camouflaged against detection, such as by stretching slowly over time or using multiple source addresses) [4][6][9]. Only with the full-scale detection of surveillance activities can security systems be augmented to match this arms race, organizing the flood of detected surveillance attempts with watch lists, correlation and intelligence profiling.

Full-scale surveillance detection, i.e., detecting this range of surveillance activities with high precision, presents a series of technical challenges. For example, real-time tracking of all prospective scanners within a high bandwidth network presents challenges with respect to memory use and speed, given the temporal analyses necessary to detect increasingly prevalent and stealthy scanning. Moreover, certain network tap points suffer from crippling information loss, such as the partial information accessible at a peering center due to unpredictably asymmetric routing.

This paper introduces System Detection (SysD)'s surveillance detection solution, which employs a *cascading filter* design that coordinates a series of specialized heuristics across extrapolated *connection records*, individual probes, scans and coordinated scanning groups. This design provides scalability via data reduction across aggregate filters, and detects scans and probes in high-bandwidth environments with high coverage and a low false positive (FP) rate. Two variations specialize over environment class: enclave surveillance detection (ESD) and peering center surveillance detection (PSD).

ESD is implemented and fully operational as a module in SysD's Hawkeye Operational Platform (HOP). The HOP infrastructure includes tools and APIs that allow various intrusion detection components to be "plugged in" and deployed. Hawkeye modules embody analysis algorithms (e.g. ESD), and feature extraction and data parsing procedures.

The rest of this paper is organized as follows: in Section 2, we present related work in surveillance detection. In Section 3, we describe how ESD and PSD work. Section 4 presents a detailed evaluation applying ESD over data from a large enclave at a multinational, very security conscious, client. Section 5 presents a detailed evaluation applying PSD over data from a peering point at a multinational, very security conscious, ISP. The profiling and trends analyses in Section 4 and Section 5 provide critical information for security analysts to understand surveillance activity behavior in general and to select ESD and PSD parameter settings to optimize detection. In Section 6, we offer concluding remarks and share directions of future work.

2. Related Work

Nearly all commercially available and open source intrusion detection systems contain techniques for detecting portscans and other forms of surveillance activity. The most typical approach to surveillance detection is to look for *X* "events of interest" in a rolling window of *Y* seconds [4][6].

One body of work related to ESD and PSD is that of Silicon Defense in their SPADE/SPICE project [9]. SPADE/SPICE is implemented as a SNORT pre-processing engine [8]. This allows it to be run in conjunction with other SNORT filters or solely focused on scan detection. Their approach is to collect statistically anomalous events (also cf. [1]) over long time intervals and then cluster anomalous packets together, theoretically into distinct surveillance attempts. ESD and PSD's approach, described below in Section 3, is to only consider failed connections and other specific types of connections to be evidence of surveillance attempts; ESD and PSD do not use statistical anomaly detection to detect individual probes. The primary reason for this is that, as evidenced in this paper, surveillance is so prevalent that individual probes are unlikely to be statistically anomalous in nature. ESD and PSD also differ from SPADE/SPICE in that, instead of clustering to aggregate probes into scans, simple thresholding is employed, which may cast a wider net, and, demonstrably, is sufficiently precise.

Silicon Defense's paper provides additional, extensive background into the problem of detecting surveillance activity, surveying existing and past systems.

Another system of note is *scanlogd*, which is a host-based scan detection program, although it can also be configured to detect portscans as a promiscuous network sniffer [7]. This program uses the canonical "X events in Y seconds" approach, but is limited to fully connected sockets and specific types of crafted packets.

3. How SysD's Surveillance Detection Works

Surveillance detection poses many difficult challenges. First, probing and scanning methods are varied, irregular and unpredictable, following unknown attacker agendas and heuristics. Second, complete session re-assembly of raw TCP/IP packet data is impractical, since it would require complete and precise knowledge of all stack implementation idiosyncrasies, local configurations across all hosts, and which packets faced network errors that prevented them from reaching their destination. Moreover, in an on-line deployment at a promiscuous network tap point, it is difficult to keep up with real-time session assembly of many connections for many hosts. Finally, stealthy surveillance may be arbitrarily spread over long time-spans, effectively hiding its behavior amongst megabytes of data.

Both ESD and PSD overcome these challenges with a three-staged process of cascading filters, as illustrated in Figure 1. First, approximate sessions between source/destination IP pairs are extrapolated with a model that is largely a simplification of that described by Lee [5]. Second, each extrapolated session that represents a failed connection attempt or otherwise "interesting" packet is assumed to be a probe. In the case of ESD, this is largely connections that show data only moving in one direction, i.e., from the source to the target, with no response.[1] Third, each probing IP is given a score based on the number of unique destination IP/port pairs probed. The IP is in turn considered a scanner if its score is greater than an empirically derived alert threshold. IPs that pass such a threshold are likely true scanners (i.e., unlikely to be false positives) since a spread of failed connections over multiple destinations has few explanations beyond purposeful scanning. The results shown in Section 4 illustrate how the alert threshold can be determined by the analyst, depending on network behavior and security concerns.

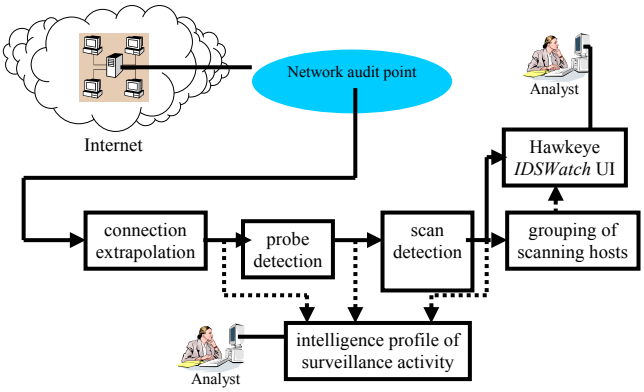

Figure 1 Pipeline architecture of surveillance detection. Dotted lines indicate optional data flow.

The primary difference between ESD and PSD is that PSD contains additional heuristics for extrapolating sessions and for probe detection when there is the possibility of asymmetric traffic (and thus visibility of communications in one direction only).

[1] When applied over archived data, as in the experiments reported in this paper, ESD discounts apparently bad connections if valid connections also exist from the same source IP to the same destination IP/port pair. This further decreases the false detection of individual probes.

Because of the precision this process provides for selecting probes by both ESD and PSD, individual probe activity may be stored for greater periods of time in order to assist in detecting long-term scanning activity. Throughout the course of our experimentation, the longest of which was over 3 days of data, there was no need to expire any records describing individual probes.

In order to reduce the number of alerts generated, multiple scanners can be grouped together if they are sufficiently close to one another in IP space (numerically, treating an IP as a literal 4 byte number). This helps find scanners who have, for example, obtained ranges of addresses from their ISP and are executing a distributed scan, or have dialed multiple times into their ISP getting a unique IP address for each connection. Such groups are considered to be under the control of the same surveillance effort. Section 4.5.3 explores the effects of grouping.

This detection algorithm is limited to the detection of surveillance that is intended to discover resources, such as canonical port-scans, rather than surveillance that intends to discover information about known resources by making valid connections. As another limitation, the algorithm could be susceptible to decoys intended to cause false positives.

4. ESD Evaluation and Alerts Analyses

ESD was applied to 74 hours of logged traffic data from a large enclave at a multinational, very security conscious, client. This data set contains 344 million packets (24 GB of headers). The average bandwidth during the capture of this data was 0.7 megabytes per second, with a 3.3 megabyte per second peak (as measured over a one second sliding window). A total of 500,000 unique IP addresses were seen.

Within the enclave data, a strikingly high percentage of attempted connections represent surveillance activity. An estimated 11.5 million connections were extrapolated by ESD, of which 13% were determined to be probes. This high percentage includes automatic probing events launched by hosts that were infected by worms (Code Red approximately 47% and NIMDA 3%), although the data was not collected during any large worm attack such as Code Red.[2]

SysD's surveillance detection solution can operate in real-time speed. Our implementation of the detection algorithm in C processes archive data 300 to 500 times the observed wire speed of the large enclave (the 74 hours of enclave data were analyzed in 14 minutes).

4.1. Detection Precision Over Test Data

A manual analysis of the alerts generated by ESD was performed to help determine how many surveillance alerts were correctly and incorrectly generated. From the unfiltered 97,152 alerts representing potential scanners, 1000 alerts (1%) were selected to undergo human analysis.

From scanners who sent between 1 probing connection and 35 probing connections, 10 alerts were randomly chosen from each level (i.e., 10 alerts were selected from scanners who sent 1 probing connection, 10 alerts were selected from scanners who send 2 probing connections, etc), for a total of 350 alerts. Additionally, 10 alerts were selected from each level from 36 to 652,118, which created a set of 1933 alerts, from which 650 alerts were randomly selected. We selected the alerts in this fashion for two purposes: we posited that as an alert threshold over the number of probes was decreased, the false positive rate would increase and thus wanted fine granularity in the range of potentially likely threshold settings; and, simple random selection would be very strongly biased towards alerts where the scanner sent probes to a low number of IPs, since such scanners are much more prevalent (as indicated by the shape of the curve in Figure 2).

ESD performs with high precision. From the set of 1000 alerts, we discovered 6 alerts that we determined were false positives (0.6% of the alerts).[3] We also characterized the reason for the generation of these alerts. The reasons were evenly split, with three alerts being generated due to local DNS servers attempting to resolve hosts on multiple lame DNS servers (DNS servers which had DNS domains delegated to them, but were either not up or not running DNS servers), and three alerts being generated due to buggy TCP/IP stacks (servers which accepted a connection, properly tore the connection down, and then an hour or two later, generated spontaneous TCP reset (RST) packets to abort this connection which had long since been shut down). We believe that both types of false positives should be easy to eliminate either through parameter adjustment, or through automated alert post-processing. (Alternately, the alerts could be revised and sent to the appropriate systems administration staff so that the necessary configuration could be corrected, or the non-

[2] The data was collected approximately six months after the initial release of Code Red.

[3] The 994 true positives inspected showed characteristics conclusively indicative of valid behavior. Our full results were given to three other parties with vested interests, and no error was discovered in our analysis.

compliant OS could be patched. However, we recognize that this may not be practical, as useful as it might be.) Due to the small sample size of false positives, no particular conclusions could be reached as to an appropriate threshold setting which would limit false positives.

This high precision shows that, in general, even scanners detected only as initiating bad connections to a small number of destination points are in fact true positives. Note that, indeed, certain other explanations for multiple, unidirectional connection destinations such as multicasting have been informally ruled out. In general, the high precision provided by ESD is possible because it is able to gain almost complete knowledge about the state of the connections it observed due to the symmetric nature of its installation point. Asymmetric traffic views require PSD.

4.2. Comparison to Production Sensor

We were given access to the alerts generated by a sensor in production use by our customer during the time our data set was gathered in order to compare them with our alerts. At equivalent threshold settings, our system found 8,603 scanners and the production sensor found 23 scanners. Their system had no false positives, and found one scanner we did not (caused by a bizarre configuration at the enclave involving a very large number of IP addresses performing honeypot decoying which hid the scanning activity from our algorithm). Our system therefore found close to 8600 scanners that the production system did not, including eight of the ten most active sources of scanning activity.

4.3. Alert Distribution Analyses

During manual analysis of the entire alert set, it was discovered that scanners were coming from 181 countries. Since there are only around 192 countries in the world, this is close or equal to every Internet-connected country. Additionally, manual analysis of the selected alerts revealed that at least 47% of the scans were generated by Code Red and 3% were generated by NIMDA. Considering Code Red had been out for approximately 10 months at the time the data set was gathered, and NIMDA had been out for 7 months, this speaks poorly of the promptness of worldwide patch installation [2][3].

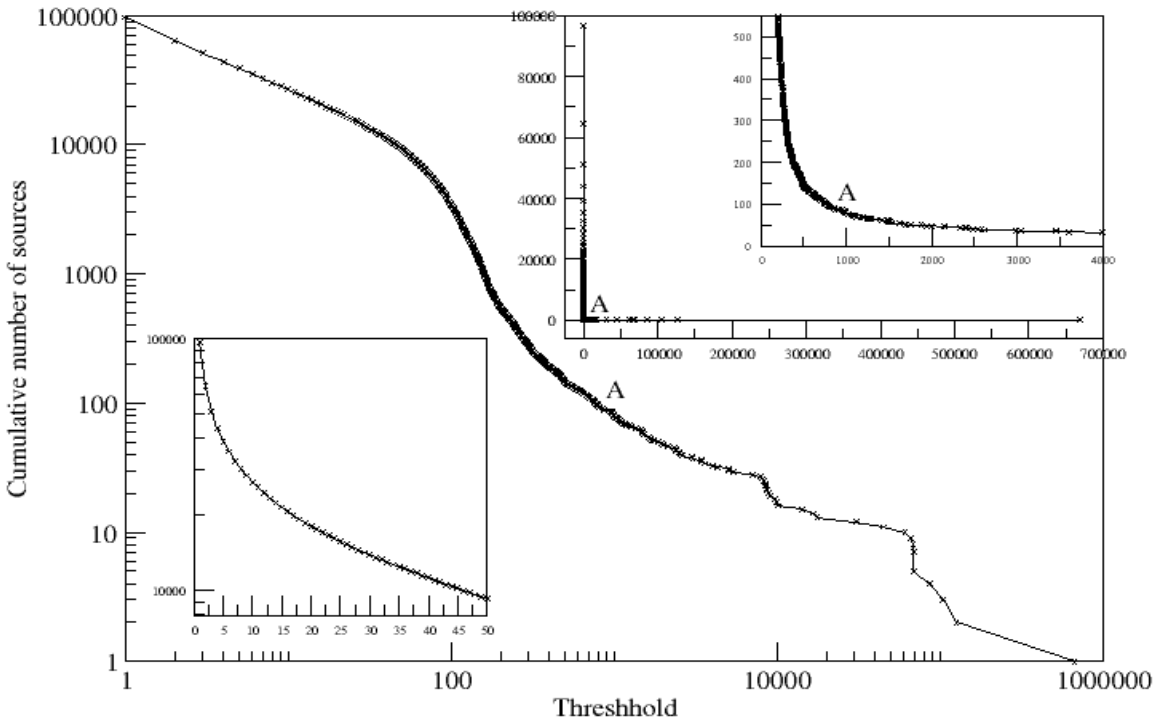

Figure 2 Number of probing sources detected by threshold. All four graphs are representations of the same data. The vertical axis of the lower left graph and both axes of the outer large graph are on a logarithmic scale.

4.4. The effect of alert threshold selection

The example analyses performed in this and the following subsection over detected surveillance activity serve to:

- Illustrate detection performance over various parameter values.
- Reveal characteristics of today's network and attack behavior, demonstrating a means of profiling different scan behaviors.
- Enable analysts to select detection parameter values appropriate for the specific network behavior and security concerns at the enclave.

As described in Section 3, the alert threshold controls the number of surveillance alerts produced; only source IPs that perform enough probes to cross the threshold will be considered a *scanner*. Threshold selection is critical for system optimization since a high threshold may result in many scans going undetected, while a low threshold may result in an overwhelming number of alerts.

Figure 2, "Number of Probing Sources Detected by Threshold," displays four graphs, each representing various views on the same data: given the choice of an alert threshold (horizontal axis), how many source IPs would be considered a scanner (vertical axis). As an example, examining Point A in the graph inset at the upper-right reveals that when the alert threshold is set at 1000, there are 79 source IPs that trigger an alert (i.e., 79 scanners).

The number of alerts can be drastically lowered with relatively small alert thresholds. As shown in the two upper-right inset graphs, the data points "hug the walls." This means that most IPs that emit probes do so at most a small number of times, and that therefore even relatively low threshold settings will eliminate all the infrequent probers, and therefore only alert on a small fraction of the sources that probe. The details of this dramatic influence of threshold selection are displayed in the upper-right inset graph.

The ability to exclude the abundance of probing IPs with a low threshold is a beneficial, positive result. This means the number of detection alerts displayed for the human analyst can be controllably low by dismissing scanners arguably less likely to be threatening in intent or focus. One motivation for dismissing IP addresses that appear to probe with very low frequency is that they are often exhibiting "backscatter" (i.e., replies sent to the enclave generated in response to packets sourced by attackers who spoof IP addresses of the enclave). Further, as mentioned above, scanners that direct probes to a wider range of target IPs are more likely to be true scanners, as opposed to an IP simply initiating a small number of bad connections for other reasons. Another reason that it may be pragmatic to ignore these relatively inactive hosts is that, if they are scanning a lot, their individual probes must be dispersed across a range of addresses beyond the protected enclave, and therefore not targeting the enclave that ESD is protecting. Note, however, that thresholding will by definition reduce the detection of the more stealthy scanners. Alternatively, by integrating multiple sensors and correlating slow or low volume alerts among a number of collaborating and distributed sites, one may distinguish backscatter and harmless events from intentional stealthy scan activity. We further address this in Section 6, below.

Because threshold selection has such a large effect in its lower ranges, the two "normal" graphs examined so far make it difficult to characterize its effects beyond the low-end of possible threshold values. This difficulty is addressed with the views presented by the other two graphs in this figure (the large one and the lower-left inset), which display the data on a logarithmic scale. This means that the values increase exponentially along each axis, revealing the effects of threshold selection across a much larger range of possible values.

The alert threshold has a particularly drastic (i.e., beneficial) effect in the very low range of values. The lower-left inset, which presents a logarithmic view zooming in on the area where thresholds may typically be set (0-50 range), reveals an acceleration of detected scanners starting in the 20-25 range. The acceleration is even greater (even in the log scale) as you approach zero. The logarithmic view reveals a number of informative "inflection points" at certain threshold values. As shown in the large graph, there are two pronounced "knees" out towards the right. This means that, at each of these areas, a relatively small increase in threshold value will eliminate a relatively large number of scanners. One interpretation of this is that, for some reason, it is unusual for an IP to surpass that amount of probe or probe-like activity. Specifically, the locations of these knees are at 8192 (8K) probes and 65536 (64K is both the size of a "class B" network and the number of ports in a TCP/IP stack) probes, which are typical configuration values to use. Also, for example, there may be an absolute limit to the probing speed of many hosts due to common network capabilities limitations or common surveillance tool configurations – recall that all the counts in this data are over only three days of activity. Speed limitations are also visible in the analysis of intraprobe delay in Section 4.5.2.

4.5. The effects of other parameters

Figure 3, Figure 4 and Figure 5 display the number of detected scanners (i.e., probing IPs) as influenced by additional detection parameters.

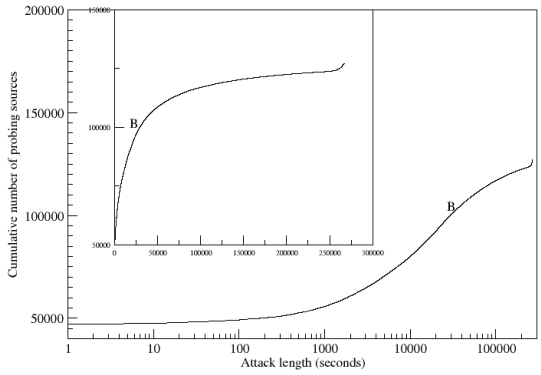

Figure 3 Number of probing sources by scan length. The inset graph is a logarithmic representation of the same data.

4.5.1. Scan Length. *Scan length* is defined as the total duration of the surveillance attempt (i.e. the time between the first and last probes). Figure 3, "Number of Probing Sources by Scan Length", displays the cumulative number of scanners for each scan length. For example, Point B indicates that 100,000 source IPs probed for 29,242 seconds or less. This is the majority of scanners – most scanners don't scan for longer than that within the 3 days of data. However, 48% of scanners spend more than one hour scanning, and 39% of scanners spend more than one day scanning. The sharp acceleration (even in the log view) near the end of the graph, for scans which lasted close to three days, is most likely not a sign that scanners only have the patience to wait three days, but instead indicates that a large number of scanners appeared to terminate simply because of the limited (three day) amount of data we received. Instead, we expect that the slowly climbing line would extend itself far to the right, indicating that some scanners spend enormous amounts of time scanning this enclave.

4.5.2. Average Intraprobe Delay. *Average intraprobe delay* is defined as the average time between probes during the duration of surveillance, i.e., from the time of a scanning IP's first probe until its last. Figure 4, "Number of probing sources by average intraprobe delay", displays the cumulative number of scanners for each intraprobe delay. That is, only probing IPs for which the average amount of time between probes is less than or equal to a given value along the horizontal axis are counted as a scanner. For example, Point C indicates that 66,814 source IPs have an average delay of 1,000 seconds or less. The majority (87%) of scanners has an average delay of 5,000 seconds or less – most scanners don't wait longer than that, on average, between probes. That is, only a small number of scanners make their scanning stealthier than that. However, 18% of scanners delay more than one hour between packets, 10% of scanners delay more than two hours between packets, 5% delay more than four hours, and 90 (0.1%) delay more than one day. Finally we discovered a very small number of scanners that waited more than 70 hours between probes, indicating a strong desire to remain unnoticed by security analysts.

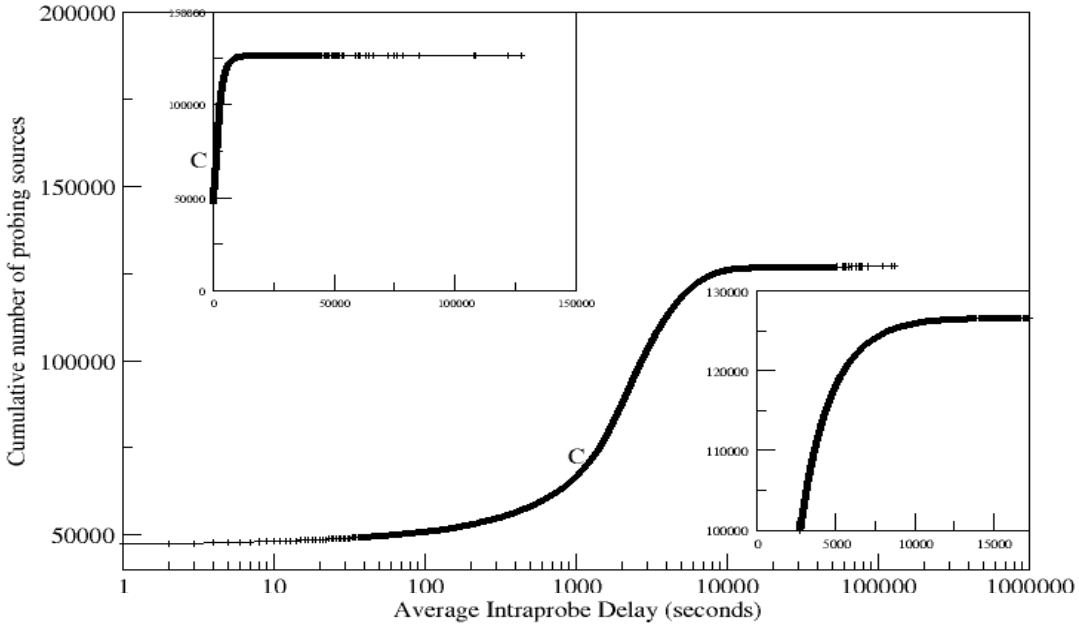

Figure 4 Number of probing sources by average intraprobe delay – three views of the same data..

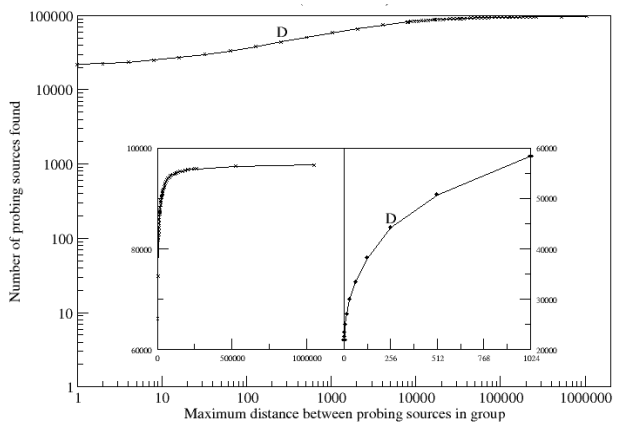

Figure 5 Number of probing sources by group distance, with a probing threshold set at 14. The outer graph is in a log scale. The inset graphs represent the same data in a normal scale.

4.5.3. Group Distance. *Group distance* is the maximum distance two IP addresses may be away from each other (numerically, treating an IP as a literal 4 byte number) in order for them to be considered part of the same surveillance effort. This produces a larger number of probing sources since nearby scanners who individually do not produce enough traffic to exceed an alert threshold (14 in this case[4]) will exceed the threshold when grouped together. Figure 5, "Number of Probing Sources by Group Distance", displays the number of scanners for each of a range of group distances, cumulatively. Given a group distance (the horizontal axis), any group of IPs that performs more than 14 probes is considered entirely composed of scanners. The vertical axis represents the aggregate number of probing IPs across all scanning groups. For example, Point D indicates that 44,179 source IPs are counted as scanners when the distance threshold for grouping is 256. The graphs show that relatively small group sizes very quickly include almost all probers (as many as are considered scanners with a threshold of 0 and no grouping) - almost all probers get indiscriminately grouped. Therefore, only small group sizes can be effective, if at all.

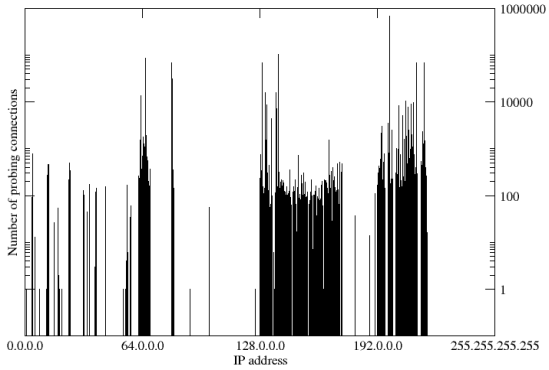

Figure 6 Number of individual probes by source IP address.

4.5.4. Additional Analyses. Figure 6, Figure 7 and Figure 8 provide additional analyses. Figure 6 and Figure 7, which are complementary, illustrate that there are fewer source hosts than target hosts, and that each source host scans more frequently than each target host is scanned. This is to be expected if there are more "good guys" than "bad guys." The wide target distribution also indicates that this enclave has a large number of IP addresses assigned to it, and that the enclave was in fact itself the source of scanning activity. Figure 8 displays the number of IP addresses seen in the dataset. While this is related to the distribution of source and target hosts, it is also interesting to note the acceleration of IP addresses seen at 10 hours, 56 hours, and 70 hours. This seems likely to be caused by address spoofing in an attempt to hide the wheat in the chaff. Backscatter from address spoofing could be a possibility, but manual analysis tends to show that the packets which were received do not correspond to this thesis. These points of acceleration are worthy of further analysis.

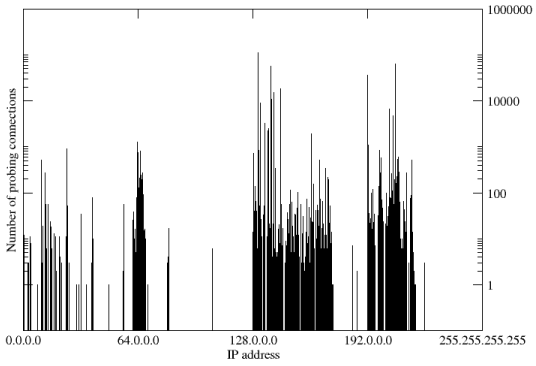

Figure 7 Number of individual probes by target IP address.

[4] The value 14 is adopted from an expert analyst at the client site who established it as a heuristic.

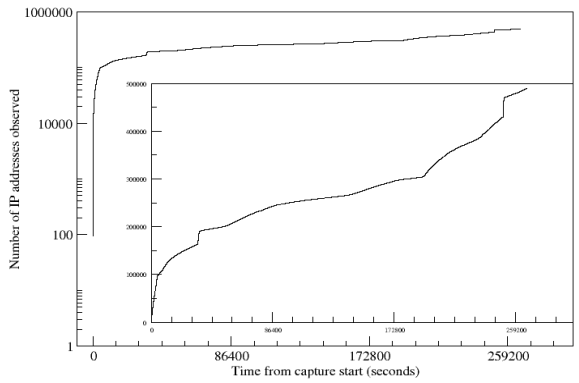

Figure 8 Number of IP address observed by time.

5. PSD Evaluation and Results

PSD was applied to 5 hours of logged traffic data from a peering point at a large, multinational, very security conscious, ISP. This data set contains 110 million packets (8 GB of headers). The average bandwidth during the capture of this data was 2.6 megabytes per second, with a 5.7 megabyte per second peak (as measured over a one second sliding window). A total of 81,000 unique IP addresses were seen. An estimated 5.4 million connections were extrapolated by PSD, of which 8% were determined to be probes. As with the data analyzed with ESD, it is probable that hosts that were infected by worms were performing much of the probing activity.

A manual analysis of the alerts generated by PSD was performed to help determine how many surveillance alerts were correctly and incorrectly generated. From the filtered list of 2,670 alerts representing potential scanners who sent fourteen or more probing connections, 652 (24%) alerts were selected to undergo human analysis. The sample of alerts which were manually analyzed was gathered from all alerts generated from traffic during one of the five hours, plus all alerts from the twenty most active scanning sources.

PSD performs with reasonable precision, depending on security task and needs. From the set of 652 alerts, we discovered 95 (14.6%) which we believed were false positives. This investigative procedure was far more difficult than what we encountered during ESD testing, as the asymmetric nature of the dataset (30% of the connections involved traffic for which we only saw one of the two participants' packet transmissions), which caused even experienced human analysis great difficulty in deciding whether or not an alert was valid. While this percentage of false positives is high, it is solely a result of the asymmetric nature of the traffic. Additionally, when considering only alerts which involved sources outside the range of addresses administered by this particular environment, the false positive rate fell to 4%. It is hoped that further research and characterization of the data set and results will provide more opportunities for false positive reduction.

While the precision of this technique is less than that of ESD, it provides a valuable tool for detection of surveillance activity at a point in the network hierarchy on the Internet backbone. The precision degradation is primarily caused by the asymmetric nature of the traffic flow. Even highly experienced human analysts looking at the asymmetric data frequently do not have enough information to adequately determine whether or not a particular packet trace represents a scan or not.

We were given access to the alerts generated by the installed system at the peering site, but we quickly discovered that this system simply produced a list of every single source which produced connections to *N* or more unique destinations over a given time window. This obviously included every scan we detected, and some 12,000 other perfectly innocuous hosts. Human analysts at the site were responsible for manually discovering the scans from this candidate pool.

A C implementation of PSD is estimated to perform with the same speed as ESD (current results were gathered using a Perl prototype).

6. Conclusions and Future Work

Both ESD and PSD have proven successful at detecting the wide range of surveillance activity within large, realistic data archives; ESD has done so with a particularly high degree of precision, and hundreds of times faster than observed wire speed.

The example analyses performed in this paper over detected surveillance activity serve to:

- Illustrate detection performance over various parameter values.
- Reveal characteristics of today's network and scanning behavior.
- Enable analysts to select parameter values appropriate for the specific network behavior and security concerns at the enclave.

These analyses lead to the following conclusions:

- Surveillance activity is very prevalent.
- Since even a low detection threshold can drastically reduce alerts, fewer false negatives may be generated.
- Grouping scanners reduces alerts, but must be limited in range to remain effective.

The cascading filter design of our surveillance detection algorithm enhances its extensibility, with the expectation that future versions must scale and evolve to match increasing bandwidths and to match the continued

arms race of surveillance methods. For example, the detection filters can be deployed in a parallel pipelined fashion for a linear increase in execution speed. Additionally, improved, specialized or refined detection techniques and heuristics can be integrated at each level by modifying the corresponding filter, matching specific advances in the arms race.

Future deployments of this technology will display, through the Hawkeye IDSWatch UI, complete intelligence profiles of surveillance activities, including graphical trend depictions such as those shown above in Section 4 for ESD. This profile consists of a range of measures and statistics that characterize surveillance trends, such as the number of scans per time unit, the number of scanners, the percentage of activity that is surveillance, the breakdown of source country frequencies, the most frequently targeted IPs, the breakdown of surveillance and surveillance-related activities (i.e., probes, scans, etc.), and temporal frequency trends (e.g., "stealthiness") of individual scanners.

Future enhancements and research will also include:

- Sharing and cross-referenced correlation of surveillance detection across distributed sites. Evidence of malicious or electronically captured probing source IPs detected by ESD and PSD installations will be shared in order to increase detection confidence, determine target ranges and distributions, and create a global view.
- Divide scanning analysis data by originating countries or infected domain hosts controlled by a client or affiliates, and visually represent it in graphical or geographical formats, i.e., a "Big Board" alert display. This process was performed manually for the example data sets.
- Inference of quantity of undetected stealthy scans, i.e., scanning activity too slow to be detected within a given time span of analyzed network behavior. This inference will be based on the falloff rate of, for example, average intra-probe delay.
- Efforts to aggregate and correlate alert streams in order to reduce analyst workload and provide intelligence discovery that reveals alert trends and enemy assessment and fingerprinting. Efforts include:
  - "Watchlist" generation of frequent offenders. As surveillance alerts are generated over days, weeks, and months, it is expected (and human analysts interviewed agree) that trends quickly emerge as to certain sources which scan continually. Additional analysis modules which score scanners and attackers by their long-term scanning (or indeed, attack) patterns will create a database of addresses which bear additional scrutiny or indeed may be passed to law enforcement for traditional prosecution (though this is frequently difficult for foreign sourced scans and attacks).
  - "Danger" assessment of protected hosts. An additional analysis module will report which IPs are most frequently targeted by watchlist-tracked IPs that are considered dangerous with respect to their originating country, organization or behavior (inferred intention). Response will include reassignment of target IPs or proactive security responses to tighten security on machines that are frequently targeted
  - Automatic clustering of scanning activities according to temporal behavior patterns to categorize activity by its source hacker method, tool or script employed, or worm incarnation. This can be considered a generalized form of passive fingerprinting. Results will in turn inform intention inference.

7. References

[1] P.K. Chan and M.V. Mahoney, "Detecting Novel Attacks by Identifying Anomalous Network Packet Headers". Florida Tech. technical report CS-2001-2, 2001.

[2] CERT, "CERT® Advisory CA-2001-23 Continued Threat of the 'Code Red' Worm". http://www.cert.org/advisories/CA-2001-23.html

[3] CERT, "CERT® Advisory CA-2001-26 Nimda Worm". http://www.cert.org/advisories/CA-2001-26.html

[4] Fyodor. "The Art of Port Scanning." Phrack 51, volume 7. September 1, 1997. http://www.phrack.com/phrack/51/P51-11

[5] W. Lee and S. Stolfo. "Data Mining Approaches for Intrusion Detection". *In Proceedings of the Seventh USENIX Security Symposium (SECURITY '98)*, San Antonio, TX, January 1998.

[6] S. Northcutt. "Network Intrusion Detection: An Analyst's Handbook". New Riders, Indianapolis, 1999. pp.122-139.

[7] Openwall Project, "scanlogd: a port scan detection tool". http://www.openwall.com/scanlogd/

[8] M. Roesch. http://www.snort.org/

[9] S. Staniford, J. A. Hoagland, J. M. McAlerney, "Practical Automated Detection of Stealthy Portscans", *Seventh ACM Conference on Computer and Communications Security*, Athens, Greece, 2000.