A Brief History of Scanning

Mark Allman
ICSI
Berkeley, CA, USA
mallman@icir.org

Vern Paxson
ICSI & LBNL
Berkeley, CA, USA
vern@icir.org

Jeff Terrell
UNC-Chapel Hill
Chapel Hill, NC, USA
jsterrel@unc.edu

ABSTRACT

Incessant scanning of hosts by attackers looking for vulnerable servers has become a fact of Internet life. In this paper we present an initial study of the scanning activity observed at one site over the past 12.5 years. We study the onset of scanning in the late 1990s and its evolution in terms of characteristics such as the number of scanners, targets and probing patterns. While our study is preliminary in many ways, it provides the first longitudinal examination of a now ubiquitous Internet phenomenon.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General; C.2.3 [**Computer-Communication Networks**]: Network Operations; C.2.6 [**Computer-Communication Networks**]: Internetworking

General Terms

Measurement,Security

Keywords

Scanning,Longitudinal,Malicious Activity

1. INTRODUCTION

Many forms of Internet activity have proven highly challenging to characterize due to the network's great diversity. Because sound characterization often requires a very *broad* perspective in order to capture the full range of variation manifested, such measurement studies face deep methodological challenges for how to *(i)* acquire sufficient breadth of measurement perspectives, and *(ii)* bring coherence to the analysis of disparate data. Yet without such studies, we lack basic insights into the global network's behavior and evolution.

With regard to Internet attack activity, several remarkable studies have managed to attain global perspectives via a range of techniques. Moore's "network telescope" provides broad visibility into distant network phenomena such as flooding attacks and worm infections by leveraging the presence in such activity of randomly selected IP addresses [8, 9, 7]. Bailey and colleagues present a system based on a wide range of distributed "black hole" network blocks coupled with probe responders [2], for which they show that the distributed perspective can be vital for capturing the range of variations that scanning activity manifests [3].

Following a different approach, Yegneswaran and colleagues studied a collection of network scanning logs from the global "DShield" repository [12, 1]. The data spanned a month-long period from 2001 (including the Code Red 2 outbreak) and a 3-month period from 2002, totaling 207 million scans sent to 1.4 million destination addresses. The study examines scanning prevalence, rates, types, and address clustering, offering a unique global perspective.

In this paper we also examine the phenomenon of scanning, but from a perspective global in *time* rather than *space*. That is, the data upon which we base our study begins in 1994—during which the measured site experienced virtually no scanning activity—through 2006, long past the point at which scanning became a ubiquitous phenomenon [10]. While only from a single site, the breadth of the data is extensive: the subset (1/30th) we use for this paper spans 628 million scans sent by 2.4 million distinct IP addresses.

We emphasize that this paper reflects *preliminary* work, with many important questions (e.g., alignment of our findings with those framed in [12]) deferred for a more extensive study we are pursuing. However, we find that even our initial "scratching the surface" of the data reveals a number of interesting results.

2. DATA

The data presented in this paper comes from 12.5 years of logs of network traffic continuously collected at the border of the Lawrence Berkeley National Laboratory (LBNL) in Berkeley, CA, USA. LBNL's address space consists of two /16 networks and a small number of /24 network blocks (the exact number varied over the span of the monitoring). Data collection began on June 1 1994, and we base our preliminary study on logs up through December 23 2006,[1] or 4,581 days. The dataset consists of one-line ASCII summaries of each incoming connection observed; see § 3.1 for specifics.

For 1994-5, the connection summaries were generated using a script that processes *tcpdump* output. In 1996, we switched to the newly written Bro intrusion detection system [11], which can produce connection summaries in the same format.

In total, the dataset includes 23.4 billion connection summaries. For our analysis, we split each day in the dataset into its own file. Grappling with data of such size poses many logistical issues. For example, some of our early analysis scripts failed because they used more than 3 GB of memory when crunching a single day's worth of

[1]Data collection continues to this day. Our initial end date was arbitrary and chosen for logistical convenience.

IMC'07, October 24-26, 2007, San Diego, California, USA.

summaries. While we re-engineered our analysis to use less memory (by taking multiple passes across the data, and by sorting the data on external IP address to allow some forms of analysis to proceed without maintaining state for multiple addresses), this serves to illustrate the problems involved in analyzing the data. In fact, the biggest problem with crunching the data is simply the amount of time required. Just to count the lines in each file with *wc* takes over 4 hours.[2] Obviously, more elaborate analysis and keeping state about the traffic slow this process down much further.

To cope with this massive size, we restrict the initial exploration of the data to analysis of every 30^{th} day. This subset of the data consists of 153 days of traffic, encompassing 794 million connection summaries (3.4% of the total number of connections in the dataset).

We note that all analysis in this paper is conducted over 24 hour periods. That is, a scanner identified on one day is not assumed to be a scanner on the next day in the dataset (unless of course we independently identify it on this second day). Given that the days in the dataset are 30 days apart in real time, this should not skew our results very much. In future work we intend to analyze the full set of data, at which point we will need to re-visit the issue of when to "age out" a scanner.

While we know that glitches in the monitoring apparatus have occurred—and thus the dataset does not include *every* connection attempt made in the last 12.5 years—there have been no prolonged outages of the monitoring infrastructure. We used the timestamps in the connection summaries to check for "short" days, finding 6 days within our dataset to be more than 30 sec "short" of a full 24-hour period. In looking back through both ancillary information and logs from surrounding days, we find that while some of these days are quite "short" (the most egregious log being roughly 6 hours long), the outages represent Bro crashes and hardware problems not correlated with scanning. Therefore, we do not believe these short days bias the assessment. In addition, we looked for gaps within each day's log. We found three days where the logs exhibit 14–16 minute gaps, and confirmed that these reflect Bro crashes. Prior to the gap we do not see scanning activity, but we *do* in the logs generated immediately after Bro restarted. Therefore, it is possible that a form of scanning affected the measurement infrastructure, slightly biasing the measurements. We do note, however, that 16 minutes is about 1% of a day, and therefore the amount of bias is likely small (especially since all of these instances come from 2004-6, when scanning was incessant, as shown in § 4). We found two additional outages of 21 and 24 minutes. These again reflect Bro problems. However, we see nothing surrounding the gaps to indicate that significant scanning was taking place during these time periods.

3. WHAT IS A SCANNER?

Crucial to our entire study is a careful definition of when to consider a remote host a "scanner". We break this question into two parts. We first consider how to classify individual connections (§ 3.1). We then use the classifications of all the connection attempts from a particular remote host to classify the remote host (§ 3.2).

3.1 Connection Classification

Our dataset consists of "connection summaries" for each incoming connection. It is important to note that while the logs include all TCP traffic (which heavily dominates LBNL's traffic mix), the presence of UDP traffic is limited to only those protocols analyzed by the Bro system, which varied over the course of its development. In addition, the logs we used do not include any ICMP traffic.

These summaries contain (1) the time a connection started (reported to μsec precision), (2) the duration of the connection (also reported with μsec precision), (3) the source and destination IP addresses, (4) the number of bytes transferred in each direction, (5) the application protocol as inferred from usage of well-known ports, and (6) a "final state" entry. This state provides a succinct summary of the connection. The "SF" state indicates that the monitor observed both the three-way SYN handshake to initiate a connection and the FIN handshake to tear it down (i.e., the connection progressed in the nominal way we think of successful connections working). The "REJ" state, on the other hand, indicates that the initial SYN (from a remote peer) elicited a RST packet from the target host, indicating a rejected connection attempt. The connections in our dataset span roughly 20 state values.[3] We classify every state but one ("SH"; see below) as either "good" for successful connections, "bad" for connection attempts that do not lead to established connections, or "unknown" for states that do not indicate clearly good or bad connections.

The "SH" state indicates that the remote peer sent a SYN followed by a FIN—however, the monitor never recorded a SYN-ACK from the local peer. At first glance, this would seem to indicate a scanner that is trying to make connection attempts look as real as possible in the hopes of not triggering an alarm. However, such connections can also indicate a vantage point problem whereby the monitor is not observing outgoing traffic from some hosts. While in general the monitor placement at LBNL can observe both incoming and outgoing traffic, there were periods of time where the traffic for some LBNL hosts would partially bypass the monitor. From a measurement perspective this is clearly undesirable. However, when conducting the sort of exhaustive long-term monitoring required to produce the dataset used in this paper, these sorts of quirks are inevitable. In this case, the quirk gives rise to traffic summaries that do not clearly show the traffic as good or bad.

To cope with this ambiguous "SH" state, we introduce a heuristic based on the *service fanout* of the remote host, i.e., the number of (localIP, port) tuples the remote host at least attempted to access. When restricted to "SH" connections, if this fanout exceeds 10 then we deem *all* the "SH" connections from the given remote host as "bad". Otherwise, we classify them as "unknown".

After processing the "SH" connections, we are left with "good", "bad" and "unknown" connections. In over two-thirds of the days in our dataset, we classify fewer than 1% of the connections as "unknown", though on 5 days we observed an unknown connection prevalence of $\geq 5\%$, with the maximum being 16%. Manual examination shows that while a variety of unknown states manifest in these days, the predominant unknown state is "SH" connections from remote hosts that do not exhibit the fanout required for us to reclassify the connections as "bad". While these connections could in fact be from scanners trying to evade detection, (i) the low fanout suggests a monitor placement issue, since fanout for legitimate remote hosts is typically low (see § 3.2), and (ii) the unknown connections are generally small enough in number as to not significantly skew our overall results.

3.2 Host Classification

Using the above classifications of each connection in our dataset,

[2]This also includes uncompression time for the largest one-third the daily logs, which we store gzip-compressed.

[3]Bro currently produces 13 different states. However, over the years additional codes have been used as understanding of how to best summarize status information evolved.

we can now turn to analyzing a remote host's aggregate behavior to determine whether to classify it as a scanner or not. Our general approach is based on the notion that connection attempts that do not result in established connections represent possible scans. Of course, there are benign reasons why such attempts occur (service temporarily unavailable, misconfigurations, user errors), and also the probes from scanners will sometimes succeed in establishing connections. Thus, simply observing a single failed connection attempt should not mark a remote host as a scanner. Accordingly, we need a heuristic for analyzing a remote host's overall activity and making a judgment.

Scanners, by definition, poke around in search of services. We can quantify the notion of "poking around" in terms of a remote host's service *fanout*, i.e., the number of (localIP, port) pairs a remote host attempts to access, as discussed in the last section. However, fanout by itself does not necessarily indicate scanning by an attacker. It could instead reflect legitimate access that happens to involve a number of different local hosts or services.[4]

One potential means for distinguishing between these two possibilities builds upon the likelihood that hostile scanners will tend to fail in their connection attempts more often than legitimate hosts, because the scanners lack knowledge of where servers reside. While previous work on detecting scanners in real-time exists (e.g., using various thresholds [6] or Sequential Hypothesis Testing [4]), for our preliminary exploration we want to start from a more relaxed definition of "scanner". We do so for two reasons: (i) such that we have an opportunity to observe patterns that might differ from those detected by operationally oriented schemes that place a premium on reacting quickly in real-time, and (ii) to allow for the possibility of ambiguity in terms of not making a decision one way or the other regarding whether a host is a scanner. To this end, we first note the fairly sharp distinction between fanout as exhibited by remote hosts that only make good connections ("good hosts") versus for remote hosts that only make bad connections ("bad hosts"). (While these hosts are not themselves hard to classify, our goal is to arrive at a well-supported rule for classifying remote hosts that make both good and bad connections to local services.)

We note that only 3% of good hosts have a service fanout exceeding 3, while 24% of bad hosts do. Clearly, good hosts generally target a small number of local services while bad hosts tend to attempt to hit a larger number of services. Next, assessing the "mixed" hosts, i.e., remote hosts that make both good and bad connections, we find that often either the good service fanout or the bad service fanout outweighs the other by a large margin (matching the similar finding reported in [4], though this time seen over a much longer period of time). We therefore formulate a rule to classify "mixed hosts" as "good" or "bad" based on the ratio of the host's fanout for good connections to its fanout for bad connections. If either type of connection outweighs the other by a factor of T, then we classify the host using the more predominant type of connection. Otherwise, we leave the host unclassified.

Figure 1 shows distributions for various values of T of the percentage of mixed hosts that remain unclassified per day using the scheme outlined above. We first note that the scheme classifies over 90% of the mixed hosts regardless of threshold we used—verifying our initial belief that remote hosts make predominantly good or bad connections. In addition, the plot shows that T values of 1.01–1.5 perform nearly identically, indicating that there is a set of remote hosts with balanced good and bad service fanout that are not easily classified using this scheme.

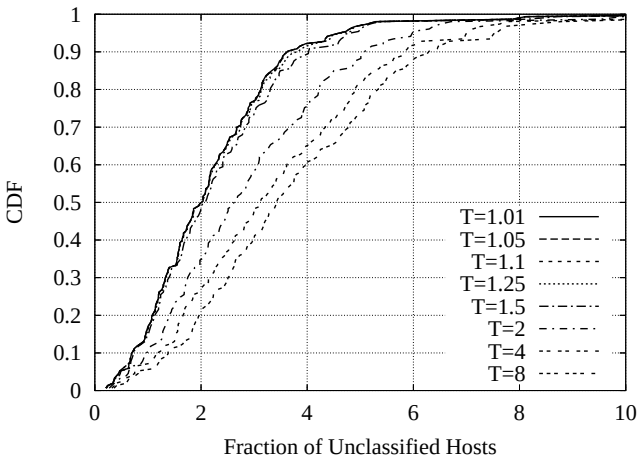


Figure 1: Unclassified mixed hosts after comparing good and bad service fanout as a function of the threshold.

Based on the above analysis, we define a scanner as a remote host that has bad service fanout of at least 4 and has bad service fanout of at least 2 times the good service fanout.

This heuristic leaves a small number of unclassified hosts (<10%); however, classifying these hosts seems difficult, at best, since they access good and bad services in roughly equal numbers. Further, to bias the results a scanner would have to engineer their scanning to interleave accesses to well known services with probes meant to figure out something about the local hosts. Such a technique has only recently been explored in the literature [5].

4. AGGREGATE VIEW

Using the above methodology for classifying remote hosts as scanners or benign, Figure 2 shows the total number of incoming connections (solid black) and the subsets from benign sources (solid gray) and scanners (dotted). The black dots indicate data from a weekend, and clarify why the data exhibits frequent oscillations: the dips nearly always occur on a weekend, when network traffic is naturally lower. The plot shows that scanning started to increase in earnest in 1998, and exhibits significant variability from month to month. In addition, 2001 emerges as marking a fundamental shift from most connections being legitimate to most being part of scanning activity, coincident with the Code Red and Nimda worm outbreaks.

Figure 3 shows the number of legitimate remote hosts and scanners attempting to establish connections each day. (We start the y-axis at 1,000 for readability, which clips the scanner host count all the way through 2000, during which it remained much smaller than the good host count.) As noted above, in terms of connection count scanning "took off" in 1998. However, in terms of number of hosts, activity only ramped up in 2001,[5] with the onset of the major worm outbreaks. From this point forward we observe thousands of scanners probing every day, and we might well consider this to reflect the onset of Internet "background radiation" [10], due to the diffuse-yet-incessant nature of scanning ever since.

The plot also shows a second, even stronger spike in early 2004, coinciding with a number of energetic scanning attacks (MyDoom,

[4]In addition, a *lack* of fanout does not necessarily mean a host is not a scanner. It might instead be scanning the Internet very broadly, such that in our dataset we only see one or two connection attempts. In principle, we might be able to assess the degree to which this occurs by cross-analyzing our dataset with a global database such as that provided by DShield [1] (at least for data from recent years).

[5]The scanner count exceeded 30/day only once prior to Oct. 1999.

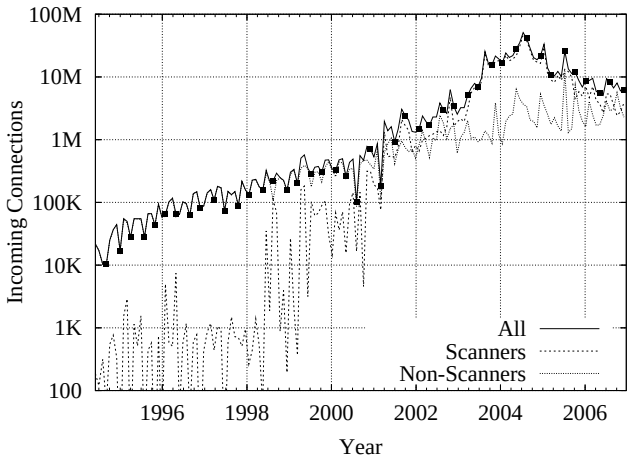


Figure 2: Connection-level summary of incoming traffic.

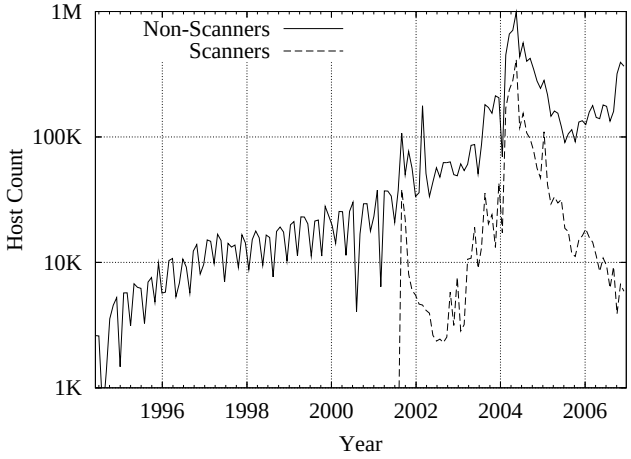


Figure 3: Host-level summary of incoming traffic.

Sasser, Welchia, Bobax, Gaobot). We note that during both spikes, we observe significant increases in the counts of non-scanners, too. This likely indicates that we are not correctly identifying all scanners. In particular, in § 3.2 we noted that 76% of hosts with no legitimate connections had fanouts ≤ 3—and hence were not labeled as "scanners". Thus, the rise in non-scanners in the Figure almost surely reflects scanners that happened to only probe LBNL's address space a small number of times.

An interesting effect is that scanning—both as a fraction of the connection attempts and as a fraction of remote hosts—has *decreased* since mid-2004. For our preliminary study, we can at this point only speculate as to likely causes. First, the peak was likely caused by very aggressive scanning worms/bots in the 2003–2004 timeframe, which inflated the scanning growth rate. Second, scanners have become increasingly refined in their techniques (cf. the discussion in § 5 of scanning rates), both for greater efficiency and to operate with a lower profile. We note that the last data point shows over two-thirds of the connection attempts in Dec. 2006 were scans, but these come from just over 1% of the remote hosts that contacted LBNL that day. Clearly, this is an area of considerable interest for our future work.

Finally, we note that in both Figure 2 and Figure 3 the non-scanning activity exhibits consistent growth of 45%/year (number of connections) and 36%/year (number of hosts), other than the aberrant elevated period during 2003–2004. This highlights both the sustained exponential growth of different dimensions of network traffic, and how scanning can occur at a level that significantly perturbs these otherwise-steady trends.

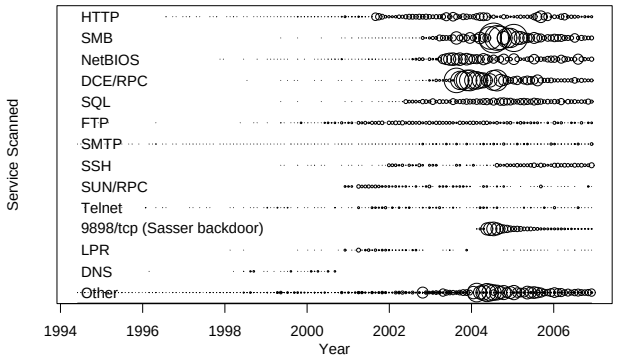


Figure 4: Services scanned as a function of time.

We next turn to what services the scans target. Figure 4 shows the services most commonly probed (greatest number of days for which they accounted for $\geq 5\%$ of all scans), with the areas of the circles proportional to the absolute number of scans for the given service on the given day. Some services see low-rate incessant scanning (e.g., SMTP), while others flare up and later die off, such as HTTP scanning in Aug. 2001 (Code Red) or the seismic onset of DCE/RPC in Aug. 2003 (Blaster), as well as SMB and NetBIOS due to other worms and bots exploiting Windows services, as noted above. For some services, we see a gradual retreat, such as for 9898/tcp, presumably reflecting on-going disinfection activities. The 9898/tcp scans are also interesting because they are *parasitic*: they reflect an attacker searching for backdoors left behind by previous malware (the Sasser worm). Finally, we note that there are occasions with heavy "other" scanning. Some of this reflects a specific constellation of scanning associated with a single worm or exploit tool, but others appear quite diverse, highlighting that a simple list of top services scanned will often not fully capture the breadth of activity.

5. SCANNER-LEVEL VIEW

We finish our brief study with a look at the behavior of individual scanners. Figure 5 shows the median and maximum number of probes sent per scanner during each day. For nearly all of the measurement period, the median number is well under 100, and since the onset of *background radiation* in 2001, the median rate has held steadily right around 10 scans/day. At the upper end, however, we see that energetic individual scanners arose directly during the 1998 onset of significant scanning activity. After that point we repeatedly observe scanners sending tens of thousands of probes, and the number that these "heavy hitters" sent has further risen over time. Also, from 1998–2001 we see somewhat of a plateau of the maximum number of scans at around 64K. We conjecture (informed by the fanout we discuss next) that this plateau comes from scanners probing a particular port on every address within one of LBNL's /16 networks. The increase after 2001 may reflect scanners that now probe both /16s, and/or an increase in scanners targeting multiple ports.

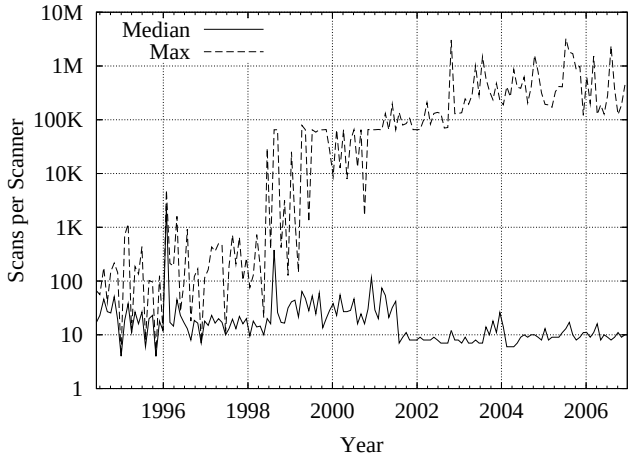
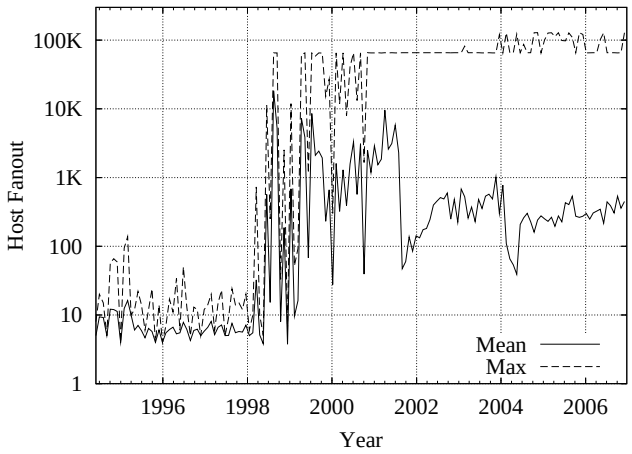


Figure 5: # scans per scanner as a function of time.

We next look at the reach of scanners in terms of how many local hosts and services they probe. Figure 6 shows host and port fanout over time. Consistently, fanout in terms of hosts is significantly higher than for ports, indicating that scanners are mainly interested in broad scans across many hosts rather than deep scanning of particular hosts ([12] notes the same effect). In addition, with the onset of significant scanning in 1998, it is not rare to observe a single scanner sweep one of LBNL's /16 networks (64K addresses), with this becoming a daily occurrence in early 2001 (not due to the worms of that year, which arrived later).

Furthermore, starting in 2004 we begin to observe single hosts scanning both of LBNL's /16 networks (128K addresses). For instance, the scanner probing the most LBNL hosts in one day over the course of our entire dataset represents a sequential probe for SQL servers on Nov. 12, 2004. The source scanned one of LBNL's entire /16 networks at 3:45AM, in just over 18 seconds, and the second one at 9:11AM. Assuming the sequential scanner probes a /16 every 18 seconds, this suggests that in the interim between the two visits to LBNL it could probe about 1,078 other /16s. In fact, there are 1,008 /16s *between* LBNL's two /16s, strongly indicating that this scanner was simply scanning a large swath of the network sequentially. This anecdote points to an important area of future research, namely assessing the patterns of scanning. Finally, we note that while in general scanners concentrate on a small number of ports, probes of over 10K ports are not uncommon in recent years.

Figure 7 shows the probing rate of scanners over time, for scanners that sent at least 100 probes. We observe a median scanning rate under one scan/sec across nearly the entire dataset. With the onset of background radiation in 2001, we also consistently see at least one scanner per day that sends 100 probes spread over the entire day (for a rate of 0.001 scans/second). Regarding the top rates, we note two sharp increases, in late 1998 and in early 2004, which we attribute to scanning software becoming more efficient, i.e., by using non-blocking calls, raw I/O, and multiple processes/threads. It is striking that prior to 1998, we never observe a scanner probing faster than 1/sec (presumably limited by a simple scanning loop that uses the OS's standard `connect` facility); and that after that point, maximum scanning rates sustained annual growth of around 170%/year. As a final note, the maximum scanning rate we observe is 88K scans/second. However, this rate comes from 137 scans over 1.5 msec. One of the many areas for future work is to assess the *sustained* scanning rates of large scanners.



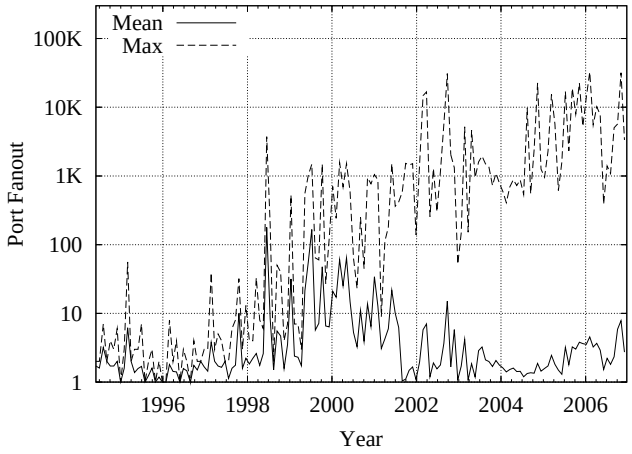


Figure 6: Fanout to local hosts (top) and ports (bottom) for each day in the dataset.

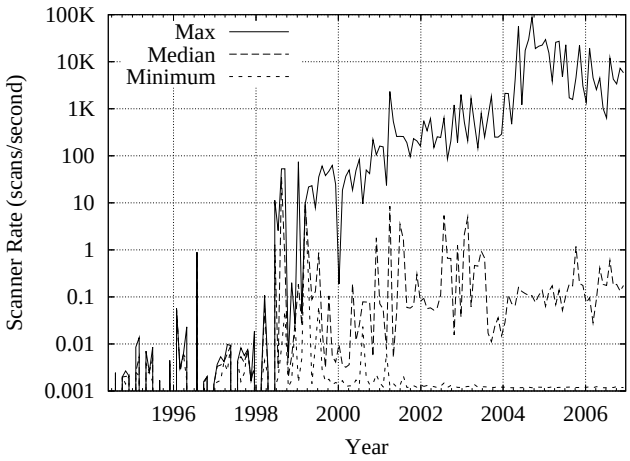


Figure 7: Scanning rate as a function of time.

6. SUMMARY

In this paper we have taken a first, brief look at the history of scanning over the last 12.5 years as observed at LBNL. While other studies have shown greater breadth in gathering data from multiple vantage points, and greater depth in the analysis of particular scanners, we are not aware of any prior work that explores the phenomenon over such an extended period of time. To be sure, there are many more questions than we have answered in this paper, which we intend to explore in future work. Some of the future data analysis we intend to pursue involves assessing (i) scanning patterns, (ii) distributed scanning whereby various hosts each scan a portion of the address/port space, (iii) correlations between scanning rates and general increases in the number of well-connected hosts and the speed of their connections and (iv) the sources of scanning by AS number or geographic region. These are merely examples and not an exhaustive list, as a goal in writing this initial paper is to solicit feedback on additional key questions to address.

7. ACKNOWLEDGMENTS

We thank the anonymous IMC reviewers for their comments and suggestions. This work was supported by National Science Foundation grants ITR/ANI-0205519, NSF-0433702, STI-0334088 and CNS-0627320, for which we are grateful. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors or originators and do not necessarily reflect the views of the National Science Foundation.

8. REFERENCES

[1] Internet storm center. http://www.dshield.org.

[2] M. Bailey, E. Cooke, F. Jahanian, J. Nazario, and D. Watson. The Internet motion sensor: A distributed blackhole monitoring system. In *Proc. NDSS*, 2005.

[3] E. Cooke, M. Bailey, Z. M. Mao, D. Watson, F. Jahanian, and D. McPherson. Toward understanding distributed blackhole placement. In *Proc. ACM CCS Workshop on Rapid Malcode (WORM)*, Oct. 2004.

[4] J. Jung, V. Paxson, A. W. Berger, and H. Balakrishnan. Fast Portscan Detection Using Sequential Hypothesis Testing. In *IEEE Symposium on Security and Privacy*, 2004.

[5] M. G. Kang, J. Caballero, and D. Song. Distributed Evasive Scan Techniques and Countermeasures. In *Proc. of Intl. Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*, June 2007.

[6] C. Leckie and R. Kotagiri. A probabilistic approach to detecting network scans. In *Proc. 8th IEEE Network Operations and Management Symposium*, Apr. 2002.

[7] D. Moore, C. Shannon, and k claffy. Code-Red: a Case Study on the Spread and Victims of an Internet Worm. In *Proc. ACM Internet Measurement Workshop*, November 2002.

[8] D. Moore, C. Shannon, G. Voelker, and S. Savage. Network telescopes. Technical report, Cooperative Association for Internet Data Analysis (CAIDA), July 2004.

[9] D. Moore, G. Voelker, and S. Savage. Interring Internet Denial-of-Service Activity. In *Proceedings of the 10th USENIX Security Symposium*. USENIX, August 2001.

[10] R. Pang, V. Yegneswaran, P. Barford, V. Paxson, and L. Peterson. Characteristics of Internet Background Radiation. In *Internet Measurement Conference*, 2004.

[11] V. Paxson. Bro: A System for Detecting Network Intruders in Real-Time. In *Proceedings of the 7th USENIX Security Symposium*, Jan. 1998.

[12] V. Yegneswaran, P. Barford, and J. Ullrich. Internet intrusions: Global characteristics and prevalence. In *Proceedings of ACM SIGMETRICS*, June 2003.