National Science Foundation
Trusted Computing (TC)
Solicitation NSF-01-160

Quantitative Network Security Analysis

David Moore, Geoffrey M. Voelker and Stefan Savage

CAIDA/SDSC and CSE Department
University of California, San Diego
9500 Gilman Drive, MS# 0505
La Jolla, CA 92092-0505

Tel: (858) 534-5160
Fax: (858) 534-5117
dmoore@caida.org
{voelker,savage}@cs.ucsd.edu

Submitted Dec 4, 2002

Contents

Quantitative Network Security Analysis

Project Summary

The field of system security research has long been dominated by individual *qualitative* results – either demonstrations of individual system vulnerabilities or expositions on the protection provided by individual security measures (e.g., firewalls, virus detectors, IDS systems, etc). These contributions, though clearly valuable, are difficult to evaluate without a complementary *quantitative* context describing the prevalence and impact of various attacks, vulnerabilities, and responses. The need for empirical data of this type is critical, both for guiding future security research and to provide a well-reasoned basis for developing operational best practices. At the same time, there are tremendous challenges in collecting and analyzing network information at sufficient scale that these findings are globally meaningful.

In previous work, we have demonstrated techniques for attacking these problems in the context of Internet-connected systems – particularly focusing on large-scale attacks such as denial-of-service and self-propagating network worms. Using a new technique, called "backscatter analysis", combined with the large address space "network telescope" we have developed at UCSD, we have been able to monitor the *global* prevalence of denial-of-service (DoS) activity on the Internet. Our approach allows us to quantitatively measure each individual attack, its duration, its intensity, and identify the victim and the services targeted. Our initial study demonstrated that DoS attacks occur with great frequency and target a wide-variety of sites and network infrastructure, thereby ending an ongoing debate in the security community about how widespread this phenomenon really was.

In related work, we have used a similar approach to monitor the spread of Internet worms such as Code-Red and Nimda. Using this data, we identified the growth pattern of these attacks, characterized the victims to identify common traits that made them vulnerable, and analyzed the effectiveness of security personnel in repairing their systems across the Internet. Finally, we have also developed a preliminary analysis of the technical requirements for effective worm countermeasures. By combining spreading models, population data extracted from real Internet worm epidemics, and measured models of Internet topology, we have shown that any reactive worm defense will require extremely widespread deployment and very short reaction times (a few minutes or less).

Using these ideas as a basis, we propose to develop a combination of network analysis techniques and network measurement infrastructure to analyze large-scale Internet security threats. In particular, we plan to investigate the following questions: how do the nature of these threats change over time, how effective are attackers at compromising services, and how well do existing security countermeasures provide a meaningful defense against these threats in practice? Using the large "network telescope" we have developed at UCSD in combination with smaller monitoring platforms on other networks, we expect to be able to measure the vast majority of large-scale Internet attacks and capture global DoS, worm, and port scan activity on an ongoing basis. Based on this longitudinal data, we will develop analytic techniques for measuring long-term trends in the make-up and staging of these attacks. We plan to extend our backscatter algorithms and measurement infrastructure to track Internet attacks in real-time and actively probe victimized hosts to understand the impact of these attacks, the distribution of various vulnerabilities, and the efficacy of employed security measures. Finally, we will modify our monitors to redirect a subset of packets to simulated hosts (a so-called "honeynet") to automatically identify and characterize new worms as they emerge.

The potential impact of this proposal is the creation of an empirical dataset that describes large-scale attacks across the global Internet. There is no equivalent dataset available today for researchers or practitioners to engineer their systems or to model the relative importance of different threats. Moreover, the real-time nature of this dataset could be widely valuable for operationally detecting, tracking, and characterizing large-scale threats as they occur. Given ongoing requests from government, industry, and academia that we receive for our preliminary data, we believe that there is keen, widespread interest for the large-scale data that we propose to create.

1 Results from Prior NSF Grants

David Moore has been involved in two NSF grants as a co-PI. The "CAIDA: Cooperative Association for Internet Data Analysis" grant recently concluded (NCR-9711092, $3,143,580, Oct 1997 - Jul 2002), although the research group, CAIDA, created under the auspices of this grant continues with additional sources of funding. CAIDA is a collaborative undertaking that brings together organizations in the commercial, government and research sectors. CAIDA provides a neutral framework to support cooperative technical endeavors and encourages the the creation and dissemination of Internet traffic metrics and measurement methodologies. Work under the CAIDA NSF grant produced over 38 published papers on Internet measurement. Moore's research under this grant has fallen primarily into the areas of topology [1, 2, 3], performance and bandwidth estimation [4, 5, 6], traffic characterization [7, 8] ([8] was fast-tracked into IEEE/ACM Transactions on Networking), tool development [9, 10, 11], and quantitative Internet security measurement[12, 13, 14]. The "backscatter analysis" paper [12] won best paper award at Usenix Security 2001, and work on the Code-Red worm [15, 14] was covered extensively by local, national and international news organizations.

As part of the CAIDA grant, Moore lead several tool development efforts. `CoralReef`[9, 10], a suite of tools for passive network measurement, has been used in numerous university courses and was the basis of an educational CDROM developed by the NSF-funded Internet Engineering Curriculum grant. `NetGeo`[11], a publically available service for mapping IP addresses to geographic locations, typically serves over 500,000 requests from over 4,000 clients per day. Both `CoralReef` and `NetGeo` have been licensed from the University of California; `NetGeo` is an integral component of a commercial geographic location service. Additionally `CoralReef` has been used under DARPA contract by SPAWAR in San Diego as part of the Navy's Reconfigurable Land Based Test Site (RLBTS) to measure one-way communication delays.

2 Introduction

Securing an individual computer host is a hard problem - as it has been for the last 20 years. Securing millions of interconnected hosts under autonomous administrative control is far more daunting, and yet that is the scope of the problem facing the Internet today. In hindsight, it is obvious that the combination of unsecured resources, unrestricted communications, and virtual anonymity makes the Internet an ideal environment for developing and targeting large-scale distributed attacks. Yet in February of 2000, few were prepared when a single attacker mustered the resources of several hundred hosts to overwhelm and effectively shut down several bellwether e-commerce sites. This was the first large-scale Internet denial-of-service (DoS) attack, a phenomenon that now occurs in excess of 3,000 times a week [16]. Eighteen months later, a different attacker released a self-propagating worm that compromised 360,000 hosts in half a day and mounted its own DoS attack against a government site [14]. Several new worms epidemics soon followed, each improving on the previous effort and some building on the "backdoors" left by previous waves. Six months later researchers described how to engineer worms that could spread orders of magnitude faster [17].

Today, it is unclear how these threats are evolving, which attacks are being deployed, how they are impacting services, or what effect current security practices are having. It is our thesis that quantitative empirical measurements of network security phenomena such as these are essential for understanding the scope of today's problems and the direction of tomorrow's, and for evaluating security technologies within an objective engineering context. Without this information, it is difficult to focus research efforts, operational practices, and policy decisions to best address these problems given the limited time and resources available.

Unfortunately, there are multiple obstacles hampering the widespread collection of such data. Generally, most individual corporations and service/content providers do not have a monitoring infrastructure that allows network security threats to be detected and tracked. Moreover, those providers that do monitor security events usually treat the data as sensitive and private. Finally, even if all organizations provided open access to their networks, monitoring and aggregating traffic from enough locations to obtain representative measures of Internet-wide behavior is a significant logistical challenge. As a result, researchers, security professionals, and policy makers must reach conclusions about the significance of various threats using a combination of intuition, anecdotal reports, and the survey data produced by organizations such as CSI and CERT.

While there is no single silver bullet for overcoming all of these challenges, we have found that there is significant leverage in harnessing the structural organization of the Internet and the interactions among its protocols. For example, we have observed that an unintended consequence of the randomly generated source addresses used in most denial-of-service attacks is "backscatter" packets that are emitted uniformly across the Internet from each victim. To each recipient these single packets appears as noise, but when they are collected in a "network telescope" and correlated across large portions of Internet address space they clearly reveal the presence of DoS attacks. Similarly, the scanning patterns used by network worms can be observed in patterns of requests to large extents of network address space. In both of these examples, we have shown how these techniques can be used to infer DoS and worm activity at a *global* scale using only *local* measurements.

In the remainder of this proposal we discuss these techniques and our preliminary findings in depth, and then outline our research goals in taking this work further. In particular, we have four specific goals we propose. First, we wish to re-engineer our prototype measurement infrastructure to provide real-time analysis about current Internet-wide security anomalies. This will allow us to drive additional active measurements to characterize the impact of attacks, the presence and effectiveness of security countermeasures and patches, and to better correlate this data with other sources of network measurement data. Second, we want to increase the sophistication of our monitoring apparatus to emulate a variety of operating systems and software platforms. This will help us detect and characterize new worms and other active scanning activity. Third, we plan to track these events on a long-term basis to extract trends in the prevalence, make-up, and staging of large-scale Internet attacks. Creating this kind of longitudinal dataset is essential for understanding the evolution of the threats and vulnerabilities being exploited. Finally, while we have validated our initial results concerning large-scale homogenous attacks, it is an open question how well these techniques can be used for also observing smaller or highly skewed attack distributions. We plan to evaluate the resolution of our techniques by comparing with smaller scale monitors and direct measurements on individual networks. Together these efforts will produce the first meaningful datasets about large-scale network attacks on the Internet. It is our hope that this data will ultimately have impact beyond the individual results that we report, and will change the way network security decisions are made.

3 Inferring Internet Denial-of-Service Activity

In this section we describe our initial work on monitoring denial-of-service activity in the global Internet. We believe that a strong quantitative approach to large-scale network security measurement is necessary both for understanding the nature of today's threat and as a baseline for the longer-term comparison and analysis research we are proposing.

Our preliminary work seeks to answer the simple question: "How prevalent are denial-of-service attacks in the Internet today?". As a means to this end, we describe a traffic monitoring technique called "backscatter analysis" for estimating the *worldwide* prevalence of denial-of-service attacks. Using backscatter analysis over a three-week period, we observe 12,805 attacks on over 5,000 distinct Internet hosts belonging to more than 2,000 distinct organizations. We further are able to estimate a lower-bound on the intensity of such attacks – some of which are in excess of 600,000 packets-per-second (pps) – and characterize the nature of the sites victimized.

In the rest of this section, we briefly describe the underlying mechanisms of denial-of-service attacks, the backscatter analysis technique we have developed, and our results from analyzing the attacks we have monitored.

3.1 Background

Denial-of-service attacks consume the resources of a remote host or network that would otherwise be used for serving legitimate users. The most damaging class of DoS attacks are flooding attacks that overwhelm a victim's CPU, memory, or network resources by sending large numbers of spurious requests. Because there is typically no simple way to distinguish the "good" requests from the "bad", it can be extremely difficult to defend against flooding attacks. Given the importance of these kinds of attacks, in our work we focus on monitoring flooding DoS attacks.

There are two related consequences to a flooding attack – the network load induced and the impact on the victim's CPU. To load the network, an attacker generally sends small packets as rapidly as possible since most network devices (both routers and network interface cards) are limited not by bandwidth but by packet processing rate. Therefore, packets-per-second are usually the best measure of network load during an attack.

An attacker often simultaneously attempts to load the victim's CPU by requiring additional processing above and beyond that required to receive a packet. For example, the best known denial-of-service attack is the "SYN flood" [18] which consists of a stream of TCP SYN packets directed to a listening TCP port at the victim. Without additional protection, even a small SYN flood can overwhelm a remote host. There are many similar attacks that exploit other code vulnerabilities including TCP ACK, NUL, RST and DATA floods, IP fragment floods, ICMP Echo Request floods, DNS Request floods, and so forth. Furthermore, attackers can (and do) mount more powerful attacks by

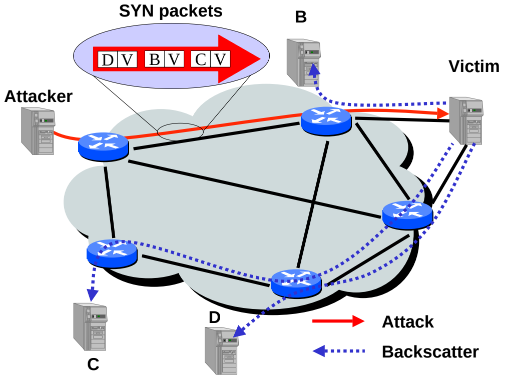


Figure 1: An illustration of backscatter in action. Here the attacker sends a series of SYN packets towards the victim using a series of random spoofed source addresses named B, C, and D. Upon receiving these packets the victim responds by sending SYN/ACKs to each host whose address was spoofed by the attacker.

combining the resources of multiple hosts in a distributed denial-of-service attack (DDoS). Our backscatter technique is able to monitor flooding DoS attacks for all such code vulnerabilities and distributed attacks.

3.2 Backscatter analysis using a network telescope

Attackers commonly spoof the source IP address field to conceal the location of the attacking host. The key observation behind our technique is that, for direct denial-of-service attacks, most programs select source addresses at random for each packet sent. These programs include all of the most popular distributed attacking tools: Shaft, TFN, TFN2k, trinoo, all variants of Stacheldraht, mstream and Trinity. When a spoofed packet arrives at the victim, the victim sends what it believes to be an appropriate response to the faked IP address.

Because the attacker's source address is randomly selected, the victim's responses are equi-probably distributed across the entire Internet address space, an inadvertent effect we call "backscatter"[1]. Figure 1 illustrates this behavior using an example of three hosts (B, C, and D) receiving backscatter packets due to one host attacking a victim.

Assuming per-packet random source addresses, reliable delivery, and one response generated for every packet in an attack, the probability of a given host on the Internet receiving at least one unsolicited response from the victim is $\frac{m}{2^{32}}$ during an attack of m packets. Similarly, if one monitors n distinct IP addresses, then the expectation of observing an attack is:

$$E(X) = \frac{nm}{2^{32}}$$

By observing a large enough address range, what we refer to as a *network telescope* [20], we can effectively "sample" all such denial-of-service activity everywhere on the Internet. Contained in these samples are the identity of the victim, information about the kind of attack, and a timestamp from which we can estimate attack duration. Moreover, given these assumptions, we can also use the average arrival rate of unsolicited responses directed at the monitored address range to estimate the actual rate of the attack being directed at the victim, as follows:

$$R \geq R' \frac{2^{32}}{n}$$

where R' is the measured average inter-arrival rate of backscatter from the victim and R is the extrapolated attack rate in packets-per-second.

3.3 Results

For our experiments we were able to monitor the sole ingress link into a lightly utilized /8 network (comprising 2^{24} distinct IP addresses, or 1/256 of the total Internet address space). We collected three traces, each roughly spanning one week, starting on February 1, 2001, and extending to February 25, 2001. Overall, we observed 12,805 attacks over the course of a week. Table 1 summarizes this data, showing more than 5,000 distinct victim IP addresses in

[1] We did not originate this term. It is borrowed from Vern Paxson who independently discovered the same backscatter effect when an attack accidentally disrupted multicast connectivity by selecting global multicast addresses as source addresses [19].

	Trace-1	Trace-2	Trace-3
Dates (2001)	Feb 01 – 08	Feb 11 – 18	Feb 18 – 25
Duration	7.5 days	6.2 days	7.1 days
Unique victim IPs	1,942	1,821	2,385
Unique victim DNS domains	750	693	876
Unique victim DNS TLDs	60	62	71
Unique victim network prefixes	1,132	1,085	1,281
Unique victim Autonomous Systems	585	575	677
Attacks	4,173	3,878	4,754
Total attack packets	50,827,217	78,234,768	62,233,762

Table 1: Summary of denial-of-service attacks in the Internet during the first three weeks of Februrary, 2001.

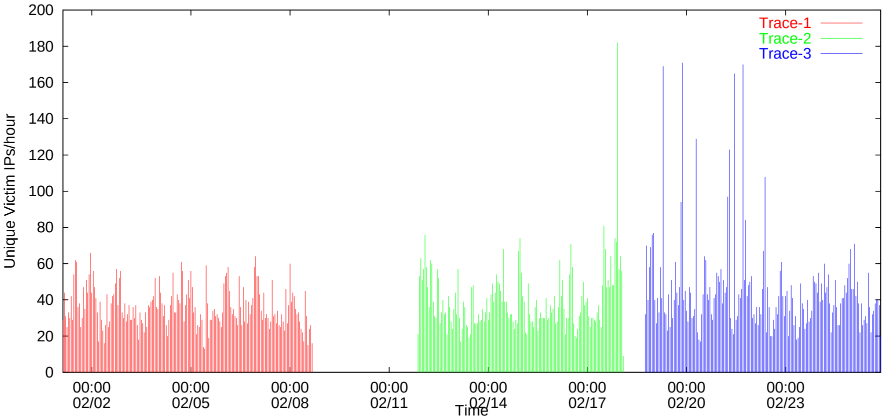


Figure 2: Estimated number of attacks per hour as a function of time (UTC).

more than 2,000 distinct DNS domains. Across the entire period we observed almost 200 million backscatter packets (again, representing less than $\frac{1}{256}$ of the actual attack traffic during this period).

In the remainder of this section we highlight results of analyses, showing attack activity over time and characterizing attacks according to their rate, duration, and their victims.

DoS attack activity over time: Figure 2 shows a time series graph of the estimated number of actively attacked victims throughout the three traces, as sampled in one hour periods. There are two gaps in this graph corresponding to the gaps between traces. The outliers on the week of February 20th, with more than 150 victim IP addresses per hour, represent broad attacks against many machines in a common network. While most of the backscatter data averages one victim IP address per network prefix per hour, the ratio climbs to above five during many of the outliers.

Attack rate: As described above, we estimate the attack rate by multiplying the average arrival rate of backscatter packets by 256 (assuming that an attack represents a random sampling across the entire address space, of which we monitor $\frac{1}{256}$). Analyzing the distributions of attack rates across all attacks in our traces, we found that 50% of all attacks have a packet rate greater than 350 packets/sec. And the most intense attack is over 679,000 packets/sec.

How threatening are the attacks that we see? Recent experiments with SYN attacks on commercial platforms show that an attack rate of only 500 SYN packets per second is enough to overwhelm a server [21]. In our traces, 46% of all attack events had an estimated rate of 500 packets/sec or greater. The same experiments show that even with a specialized firewall designed to resist SYN floods, a server can be disabled by a flood of 14,000 packets per second. In our data, 2.4% of all attack events would still compromise these attack-resistant firewalls. We conclude that the majority of the attacks that we have monitored are fast enough to overwhelm commodity solutions, and a small fraction are fast enough to overwhelm even optimized countermeasures.

Attack duration: While attack event rates characterize the intensity of attacks, they do not give insight on how long attacks are sustained. Analyzing the distribution of attack durations, we find that most attacks are relatively short: 50% of attacks are less than 10 minutes in duration, 80% are less than 30 minutes, and 90% last less than an hour. However, the tail of the distribution is long: 2% of attacks are greater than 5 hours, 1% are greater than 10 hours, and dozens spanned multiple days! Although many attacks are relatively brief, even short intense attacks can cause

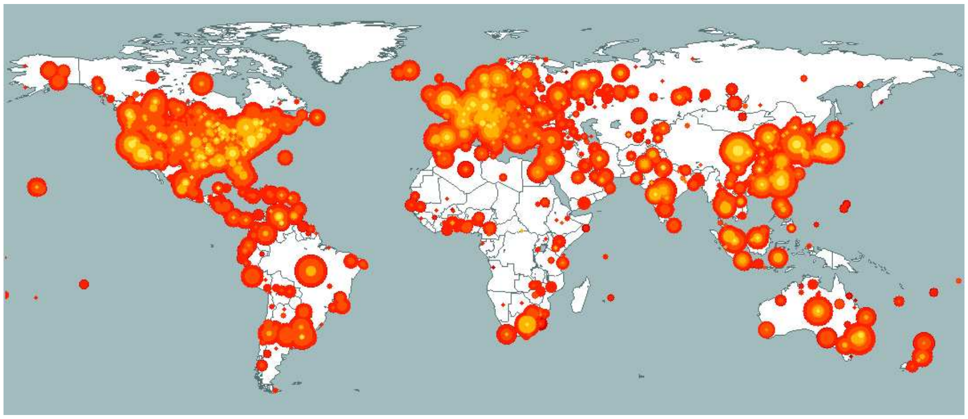


Figure 3: Geographic location and impact of the Code-Red worm.

seriously damage.

Victim characterization: Focusing on the victims of attacks, we have characterized them according to their DNS name and discovered a number of disturbing trends. First, there is a significant fraction of attacks directed against dialup and broadband home machines. Some of these attacks, particularly those directed towards cable modem users, constitute relatively large, severe attacks with rates in the thousands of packets/sec. This suggests that minor denial-of-service attacks are frequently being used to settle personal grudges. In the same vein we anecdotally observe a significant number of attacks against victims running "Internet Relay Chat" (IRC), victims supporting multi-player game use (e.g. battle.net), and victims with DNS names that are sexually suggestive or incorporate themes of drug use. We further note that many reverse DNS mappings have been clearly been compromised by attackers (e.g., DNS translations such as "is.on.the.net.illegal.ly" and "the.feds.cant.secure.their.shellz.ca").

Second, there is a small but significant fraction of attacks directed against network infrastructure. Between 2–3% of attacks target name servers (e.g., ns4.reliablehosting.com), while 1–3% target routers (e.g., core2-corel-oc48.paol.above.net). Again, some of these attacks, particularly a few destined towards routers, are comprised of a disproportionately large number of packets. This point is particularly disturbing, since overwhelming a router could deny service to *all* end hosts that rely upon that router for connectivity.

Finally, we are surprised at the diversity of different commercial attack targets. While we certainly find attacks on bellwether Internet sites including aol.com, akamai.com, amazon.com and hotmail.com, we also see attacks against a large range of smaller and medium sized businesses.

3.4 Summary

Using our "network telescope" and our backscatter analysis technique, we are able to observe global DoS activity in the Internet. Based upon our initial study, we find that DoS activity is widespread across the Internet, some are intense and long-lasting, and a surprising number of attacks target home machines and Internet services. This initial work forms the basis for the work that we are proposing, including analyzing DoS attacks over long time scales to detect long-term trends, online analysis to infer the extent of damage on the victim and whether victims instituted defenses and the efficacy of those defenses, and the impact of attacks on critical infrastructure. We discuss our proposals in more detail in Section 6.

4 Tracking the Code-Red worm

In this section, we describe how we used our "network telescope" to track the spread of the Code-Red worm throughout the Internet, and our analysis of the victims and impact of the worm.

On June 18, 2001, eEye released information about a buffer-overflow vulnerability in Microsoft's IIS web servers [22]. Microsoft released a patch for the vulnerability eight days later, on June 26, 2001 [23]. On morning of July 19th, 2001, we observed the spread of a new Internet worm dubbed Code-Red that infected hosts running unpatched versions of Microsoft's IIS web server. The worm spread by probing random IP addresses and infecting all hosts vulnerable to

DNS-based host types		
Type	Average Hosts	Hosts(%)
Unknown	88116	54.8
Other	37247	23.1
Broadband	19293	12.0
Dial-Up	14532	9.0
Web	846	0.5
Mail	731	0.5
Nameserver	184	0.1
Firewall	9	0.0
IRC	2	0.0

Table 2: The classifications of hostnames based on reverse-DNS lookups of the IP addresses of Code-Red infected hosts. Shown here are the average number of active hosts in each two hour interval and the overall percentage of each type of host across the whole seven day interval. Unknown hosts had no reverse DNS records.

the IIS exploit. Remarkably, this worm infected more than 359,000 machines across the worldwide Internet in just fourteen hours [24][25].

To illustrate both the severity and global impact of the Code-Red worm, Figure 3 shows the geographic location of hosts on the Internet infected by Code-Red after 14 hours of initial infection. Each circle on the map corresponds to a concentration of infected hosts, and the radius of the circle exponentially measures the number of infected hosts at that location.

In the rest of this section, we describe how we used our network telescope to track the Code-Red worm and our results analyzing the spread of the worm and the victims that it infected.

4.1 Methodology

Our analysis of the Code-Red worm covers the spread of the worm between July 4, 2001 and August 25, 2001. Using the same network telescope as in our DoS study (Section 3.2), we captured traces of infected hosts probing random IP addresses in our monitored network. Because of the nature of worm probe requests compared to DoS response backscatter, we were able to easily differentiate the two types of data.

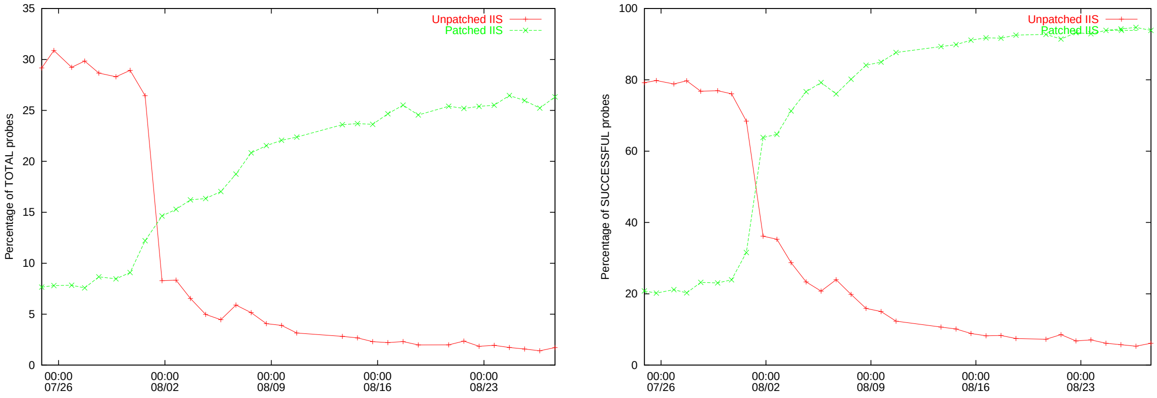
4.2 Host infection rate

We detected more than 359,000 unique IP addresses infected with the Code-Red worm between midnight UTC on July 19 and midnight UTC on July 20. To determine the rate of host infection, we recorded the time of the first attempt of each infected host to spread the worm. Because our data represent only a sample of all probes sent by infected machines, the number of hosts detected provides a lower bound on the number of hosts that have been compromised at any given time. Our analysis showed that the rate of the spread of the worm is exponential, and that the infection rate peaked at a daunting 2,000 host/minute.

4.2.1 Host Classification

We utilized the reverse DNS records for the Code-Red infected hosts to identify the function of the compromised machines. While reverse DNS records did not exist for 55% of the hosts infected, we did manage to identify about 22% of the host types. Computers without reverse DNS records are less likely to be running major services (such as those demonstrated in the other host types).

Broadband and dial-up services represented the vast majority of identifiable hosts, as shown in Table 2. Furthermore, we measured large diurnal variations in the number of infected hosts suggest that these machines are unlikely to be running production web servers of any kind, a surprising result given that the worm attacks a vulnerability in web servers. Overall, the number of broadband and dial-up users affected by this random-source worm seems to significantly exceed those affected by random-source denial-of-service attacks. While 21% of all hosts compromised by Code-Red were home and small business machines, only 13% of random-source denial-of-service attack targets shared this characteristic.



(a) All survey attempts.

(b) Hosts which responded.

Figure 4: Patching rate of IIS servers following initial Code-Red v2 outbreak on July 19th.

4.2.2 Repair rate

We performed a follow-up survey to determine the extent to which infected machines were patched in response to the Code-Red worm. Every day between July 24 and August 28, we chose ten thousand hosts at random from the 359,000 hosts infected with Code-Red on July 19 and probed them to determine the version number and whether a patch had been applied to the system. Using that information, we assessed whether they were still vulnerable to the IIS buffer overflow exploited by Code-Red.

Although this data does not show the immediate response to Code-Red, it does characterize the efficacy over time of user response to a known threat. Between July 24 and July 31, the number of patched machines increased an average of 1.5% every day. Despite unprecedented levels of local and national news coverage of the Code-Red worm and its predicted resurgence on August 1, the response to the known threat was sluggish. Only after Code-Red began to spread again on August 1 did the percentage of patched machines increase significantly, rising from 32% to 64%.

We observed a wide range in the response to Code-Red exhibited by the top ten most frequently infected top-level domains. The EDU top-level domain exhibited a much better patching response to Code-Red than did COM or NET – 81% of infected hosts were patched by August 14. COM (56%) and NET (51%) did respond well, ranked third and sixth, respectively.

4.3 Summary

The primary observation to make about the Code-Red worm is the speed at which a malicious exploit of a ubiquitous software bug can incapacitate host machines. In particular, physical and geographical boundaries are meaningless in the face of a virulent attack. In less than 14 hours, 359,104 hosts were compromised.

This assault also demonstrates that machines operated by home users or small businesses (hosts less likely to be maintained by a professional systems administrators) are integral to the robustness of the global Internet. As is the case with biologically active pathogens, vulnerable hosts can and do put everyone at risk, regardless of the significance of their role in the population.

This initial study forms the basis for the research we are proposing on Internet worms, including long-term data collection and analysis of worm spread and life cycle, using large-scale honeynets [26] to capture and study worms in detail, and containments techniques for mitigating the damaging effects of worms. We further discuss our long-term research goals for tracking Internet worms in Section 6, and in the next section we describe our preliminary work on systems for safeguarding the Internet from future catastrophic worms.

5 Internet Quarantine: Containing Self-Propagating Code

In the previous section we measured the spread and impact of the Code-Red worm. In this section, we describe our preliminary work on systems for safeguarding the Internet from worms. Unfortunately, as demonstrated by the Code-Red episode, we do not currently have an effective defense against such threats. While research in this field is nascent, traditional epidemiology suggests that the most important factors determining the spread of an infectious pathogen are the vulnerability of the population, the length of the infectious period and the rate of infection. These translate into three potential interventions to mitigate the threat of worms: *prevention, treatment, and containment*. Our initial focuses exclusively on the last approach.

Containment technologies, as exemplified by firewalls, content filters, and automated routing blacklists, can be used to block infectious communication between infected and uninfected hosts. In principal, this approach can quickly reduce, or even stop, the spread of infection, thereby mitigating the overall threat and providing additional time for more heavy-weight treatment measures to be developed and deployed. During the Code-Red epidemic, ad-hoc containment mechanisms were the primary means used to protect individual networks (e.g., by blocking inbound access to TCP port 80, or content filtering based on Code-Red specific signatures), or isolating infected hosts (e.g., by blocking the host's outbound access to TCP port 80). These solutions were implemented manually using existing routers, firewalls, and proxy servers. While these limited quarantines did not halt the worm's spread, they provided limited protection to portions of the Internet.

In our initial work, we investigate the use of widespread containment mechanisms as an approach for mitigating network-borne epidemics. However, rather than proposing particular technologies to detect or contain network worms, we have focused on a more basic question: How effectively can any containment approach counter a worm epidemic on the Internet? We consider containment systems in terms of three abstract properties: the time to detect and react, the strategy used for identifying and containing the pathogen, and the breadth and topological placement of the system's deployment. Using a vulnerable host population inferred from our measurement of the Code-Red epidemic, and an empirical Internet topology data set, we use simulation to analyze how such a worm would spread under various defenses, ranging from the existing Internet to an Internet using idealized containment technology.

In the rest of this section, we describe our model for analyzing worm propagation and worm containment systems, and then present preliminary results on the abilities and challenges for containment systems to be successful.

5.1 Modeling Worms

While computer worms represent a relatively new threat, the mathematical foundations governing the spread of infectious disease are well understood and are easily adapted to this task. In particular, worms are well described using the classic SI epidemic model which describes the growth of an infectious pathogen spread through homogenous random contacts between *Susceptible* and *Infected* individuals [27].

We can analytically describe the the proportion of infected individuals at time t as follows:

$$i(t) = \frac{e^{\beta(t-T)}}{1 + e^{\beta(t-T)}}$$

This result is well known in the public health community and has been thoroughly applied to digital pathogens as far back as 1991 [28]. To apply this result to Internet worms, the variables simply take on specific meanings. The population, N, describes the pool of Internet hosts vulnerable to the exploit used by the worm. The susceptibles, $S(t)$, are hosts that are vulnerable but not yet exploited, and the infectives, $I(t)$, are computers actively spreading the worm. Finally, the contact rate, β, can be expressed as a function of worm's probe rate r and the targeting algorithm used to select new host addresses for infection.

This function has the characteristic that for small values of t the incidence grows exponentially, until a majority of individuals are infected, at which point the incidence slows exponentially, reaching zero as all individuals are infected.

In our work, we assume that an infected host chooses targets randomly, like Code-Red v2, from the 32-bit IPv4 address space. Consequently, $\beta = r\frac{N}{2^{32}}$, since a given probe will reach a vulnerable host with probability $N/2^{32}$. Consequently, for a fixed β, N and r are inversely proportional: the spread of a worm in a population of aN vulnerable hosts sending at rate r is the same as the spread of N hosts probing at rate r/a.

5.2 Modeling Containment Systems

To understand how various containment techniques influence the spread of self-propagating code, we simulate three factors that determine the ultimate propagation of the worm:

- *Reaction time.* We define the *reaction time* of a containment system to include the time necessary for detection of malicious activity, propagation of the information to all hosts participating in the system, and the time required to activate any containment strategy once this information has been received.

- *Containment strategy.* The containment strategy refers to the particular technology used to isolate the worm from susceptible hosts. We focus on two key strategies: *address blacklisting* and *content filtering*. The former approach, similar to that used by some anti-spam systems, requires a list of IP addresses that have been identified as being infected. Packets arriving from one of these addresses are dropped when received by a member of the containment system. This strategy has the advantage that it can be implemented with today's filtering technology, does not require the worm to be identified and has a predictable effect on traffic from a given host, but it must be updated continuously to reflect newly infected hosts. The second approach requires a database of content signatures known to represent particular worms. Packets containing one of these signatures are similarly dropped when a containment system member receives one. This approach requires additional technology to characterize worm outbreaks and automatically identify appropriate content signatures. However, it has the key advantage that a single update is sufficient to describe any number of instances of a worm.

- *Deployment scenario.* In an ideal world, every node in the network would be a participating member of the containment system. However, for practical reasons this is unlikely. Instead, containment systems may be deployed at the edge of corporate networks, like firewalls, or perhaps may be implemented by Internet Service Providers (ISPs) at the access points and exchange points in their network. Moreover, it would be unreasonable to expect that even these deployments would be universal. Consequently, we examine a range of different deployment scenarios, ranging from small numbers of customer edge networks to large numbers of highly connected ISPs.

Finally, while some combinations of parameters are sufficient to stop the spread of a worm indefinitely, others simply slow its growth. To capture the impact of this latter effect, we must limit our analysis to some finite time period. In this paper, we evaluate the success of each containment system design based on the outcome occurring after 24 hours. Experimental evidence from our measurements of the Code-Red epidemic indicates that human system administrators are not able to routinely intervene in less than a 24 hour period.

5.3 Worm containment in the Internet

Given the models we have created for representing worms and containment systems, we now develop a realistic network model and evaluate the impact of partial deployments on the effectiveness of containment. We are careful to select realistic deployment scenarios, in which some fraction of customer networks implement containment at their Internet border, or some number of the highest-connectivity ISPs do the same at their exchange points.

5.3.1 Network Model

To evaluate where and how worm containment needs to be deployed in the Internet for it to be effective, we (1) develop a model of Internet connectivity among Autonomous Systems (ASes), (2) identify a representative set of vulnerable Internet hosts and the ASes in which they are located, and (3) model AS paths among all vulnerable hosts. We refer to the collection of ASes, the mapping of vulnerable hosts to ASes, and the routes among them as the *topology* on which we evaluate worm containment.

To identify the set of ASes in the Internet and their connectivity, we used the routing table for July 19, 2001 08:00 PDT from the popular Route Views service [29]. This routing table enables us to build an AS topology graph of the Internet using the 11,582 ASes contained in the table. We chose this day and time to reflect the state of Internet routing when the Code-Red worm started propagating.

For a representative set of vulnerable Internet hosts distributed across the Internet, we use the hosts infected by the Code-Red v2 worm during the initial 24 hours of propagation [15]. This set of hosts is large, well distributed throughout the Internet address space and known to represent hosts with a common vulnerability. We model the

Location	Coverage (%)	
	AS to AS Paths	IP to IP Paths
25% Customer ASes	25.0	34.0
50% Customer ASes	50.0	56.6
75% Customer ASes	75.0	74.6
Top 10 ASes	90.9	88.3
Top 20 ASes	97.0	95.0
Top 30 ASes	98.5	97.4
Top 40 ASes	99.1	98.2
Top 100 ASes	99.7	98.9
All	100.0	99.3

Table 3: Path coverage among vulnerable ASes and end hosts.

topological location of each host by placing it in its origin ASes as determined using the Route Views data mentioned previously. Note that, because some host IP addresses map to multiple origin ASes, we cannot accurately associate them with a particular origin AS and therefore remove them from consideration. As a result, we include only 338,652 vulnerable hosts distributed among 6,378 ASes when using this network model.

We model different deployment scenarios by assigning groups of ASes to the containment system. It is assumed that an AS belonging to this system can choose to filter any packet passing through one of its routers. To model which packets pass through each AS, we compute the shortest path on the graph of all AS adjacencies in the routing table. In the absence of policy, BGP will choose the shortest AS path of all equal cost options. However, we found that many pairs of ASes were connected by multiple equal-cost shortest paths (with an average of 6.3 equal-cost paths for every AS pair). We explored several different techniques to resolve such ties and observed no significant differences between them. We break ties by selecting the AS path with the greatest outdegree sum.

5.3.2 Deployment Scenarios

It is unlikely that containment systems will be universally deployed or even deployed in a majority of large customer or service provider networks. Consequently, a key question is how well these systems behave in deployment scenarios that involve a subset of these organizations.

Table 3 lists the deployment scenarios we study and the Internet path coverage they provide. We calculate path coverage as the percentage of paths from vulnerable source hosts to vulnerable destination hosts that pass through ASes included in a given deployment scenario. The first group represents participation from the customer networks contained within varying fractions of ASes selected at random (to check for bias, we selected multiple sets of such nodes and obtained qualitatively similar results). In these scenarios, content filtering firewalls are deployed at the edge of all customer networks in these ASes and worm traffic entering or leaving these networks is blocked (but not transit traffic passing through the containing AS). The second group represents deployments at major ISPs, selected according to AS outdegree in our routing table. In this scenario, content filtering is implemented in the interfaces of all exchange point routers and can filter all incoming, outgoing and transit traffic.

5.3.3 Code-Red Case Study

Figure 5 shows the effectiveness of containment for various configurations of the two deployment scenarios using the original Code-Red parameters and the content filtering containment strategy. We select a reaction time of 2 hours, which contains the worm to less than 1% of vulnerable hosts. The y-axis of the graph shows the fraction of all vulnerable hosts that become infected 24 hours after the start of worm propagation.

The bars on the left of the graph show various degrees of deployment using the Customer deployment scenario. From the graph, we see that this deployment is only partially successful at containing the worm. Even with 75% of the ASes customer networks participating, the worm still propagates to over 25% of the vulnerable hosts in 24 hours with 95% certainty.

The bars on the right of Figure 5 show various degrees of deployment using the Top ISP deployment scenario. In this scenario, the N largest ISPs block all worm probes traversing their networks, including probes on incoming,

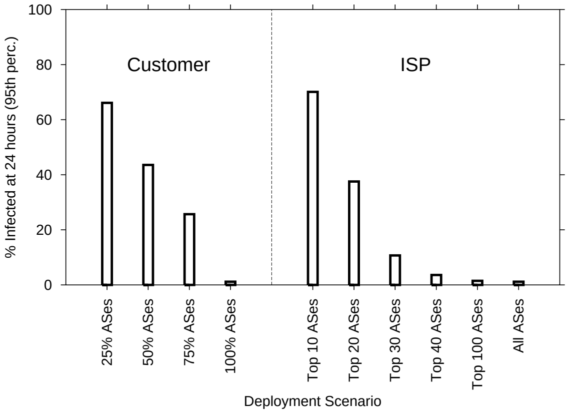
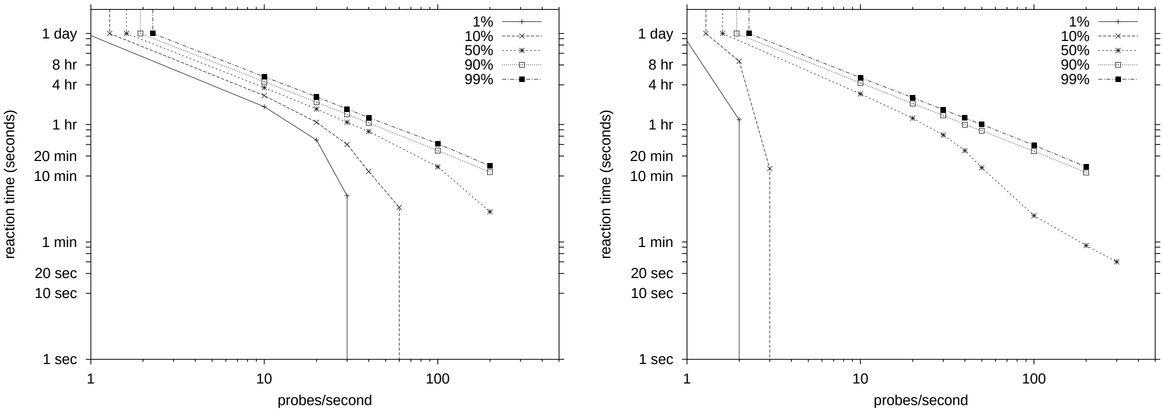


Figure 5: Containment effectiveness as a function of deployment scenario.



(a) Top 100 ISPs

(b) 50% Customers

Figure 6: The reaction times required for effective worm containment for various worm intensities. Shown are graphs for the two deployment scenarios of worm containment: (a) "Top 100 ISPs" and (b) "50% Customers."

outgoing, and transit flows. From these results, we see that a worm can be contained to a minority of hosts if the top 20 ISPs cooperate, and including the top 40 ISPs is sufficient to limit the worm to less than 5% of all hosts.

The advantages of the ISP approach relates directly to their role in carrying transit traffic. As seen in Table 3, filtering traffic at the customer networks contained within 75% of all ASes intercepts less than 75% of all potential paths. By contrast, the top 10 ISPs alone can interdict almost 90% of all paths between infected and susceptible hosts.

5.3.4 Generalized Worm Containment

We have seen the effects of more realistic deployment scenarios on the propagation of a Code-Red worm. Now we determine the reaction time requirements of these deployment scenarios on more aggressive worms by exploring the relationship between containment system reaction time and worm aggressiveness. Figure 6 shows the results for two deployment scenarios using content filtering, (a) the "Top-100 ISP" scenario and (b) the "50% Customer" scenario .

These graphs show two important results. First, we see that it is essential to model network effects: when modeling the deployment scenarios, containment encounters inherent limits. For example, when we experimented without using a network model we found that using content filtering could contain infection to 1% of vulnerable hosts with a reaction time of 2 hours. However, Figure 6 shows that neither deployment scenario can contain worms to a 1% infection at non-trivial worm probe rates. Indeed, the "Top-100 ISPs" scenario is the most effective deployment scenario, but for a worm spreading at 100 probes/second the containment system cannot prevent it from spreading to less than 18% of the hosts .

The reason why containment cannot achieve low infection rates for aggressive worms is due to the fact that the deployment scenarios do not cover all of the paths among all of the vulnerable hosts. The "Top-100 ISPs" scenario blocks 99.7% of the paths among hosts. However, at high probe rates the worm is able to infect enough vulnerable hosts before it is detected and blocked that it continues to exploit the 0.3% unblocked paths and spread further. Even with instantaneous reaction times, these 0.3% unblocked paths are enough of a backdoor that the worm can exploit those paths to infect more than 10% of the network in 24 hours.

Second, we see that the "Customer" approach to containment is again not nearly as effective as the "Top ISP" approach for low infection rates. Directly comparing the two graphs in Figure 6, the two deployment scenarios behave similarly for infection rates of 50% and above. However, for lower, more interesting infection rates, the "Customer" approach is significantly less effective. The "50% Customer" scenario cannot limit worms to 1–10% infection rates for probe rates as small as 2–3 probes/second; in other words, it cannot even limit a Code-Red worm to a 10% infection rate. Compared with the 100 largest ASes using content filtering, with the "50% Customers" scenario over 5,000 ASes are preventing the worm from infecting their own networks.

5.4 Summary

We have performed preliminary work investigating the use of widespread containment as a mechanism for mitigating network-borne epidemics. From our simulation experiments, we have found that building Internet containment systems that prevent widespread infection from worm epidemics will be very challenging. As part of the work we are proposing, we plan to investigate the plausibility of automatically detecting worm probes from within the network as well as practical systems for propagating worm probe filters among Internet routers.

6 Research Plan

1. Real-time attack detection

   While offline analyses of attack activity are useful starting points, we wish to expand our analysis to include real-time detection and characterization of attacks. The motivation for this capability is to allow additional heavy-weight measurements to be taken to further analyze individual attacks.

   For example, we wish to expand our analysis to infer the damage inflicted on the victim. The impact of an attack does not rest solely on the magnitude of the attack probe rate, but also on the victim's resilience to the attack and the number of ancillary victims who depend on services provided by the attack target. Using additional probes taken during and after an attack we can estimate the extent to which a victim's response time slows in the face of an attack and detect when a target machine hangs or crashes.

   By the same token, we also wish to detect real-time defensive intervention on the part of local or upstream network administrators and characterize the scenarios in which attacks are successfully thwarted and measure how quickly vulnerabilities are patched after an attack is activated.

   Finally, we wish to focus on the effects of denial-of-service attacks and worms on critical infrastructure, including backbone routers, DNS root servers, and the clients of these critical services. Evaluating these results will require real-time correlation with other sources of network measurement data such as routing table updates and active network latency measurements.

2. Longitudinal data analysis

   While our preliminary work in quantifying denial-of-service attacks and tracking the spread of worms has provided an initial look at the properties of these threats, we need to expand our inquiry to capture the changes and evolution in how these threats are deployed.

   We plan to collect and archive data over a period of three years to investigate trends in worm and denial-of-service activity. We will analyze trends in worm spread rates, vulnerable victim populations, repeatedly victimized computers, attack probe rates, attack distribution (including hitlists and weighting schemes), and the ratio between unique IP addresses and unique victims. Similarly, we will further examine the parameters that uniquely identify each DoS attack, including duration, probe rate, protocol, attack type (i.e. SYN flood, SMURF attack, etc), and target characteristics. After developing methods of automatic attack identification, we will seek to both quantify large denial-of-service attacks and to identify multiple attacks against a single victim.

We will also investigate long-term trends in the efficacy of host response to security threats. We will evaluate existing infrastructure to obtain a quantitative global view of intrusion risk.

Finally, we hope that such trend data will also let us develop models for describing these attacks to allow quick operational identification of new worms and DoS attacks that have not been previously seen.

3. Network telescope sensitivity

   We have demonstrated in our preliminary work that the local monitoring we employ can be used to accurately infer global large-scale activity. However, our infrastructure is unique and fixed and ideally our results could be repeated at small scale. Unfortunately, we have determined that significant bias is introduced when the effects of global events are measured on small local scales. We plan to continue our work on identifying biases introduced based on the size of the network space monitored. We will investigate methods of de-convolving wide-scale measurements into an accurate picture of the events, free from artifacts stemming from monitor scope or location.

   We will also analyze the sensitivity of our techniques to small scale network events, skewed events, or the impact of routing-based bias. To control these effects we will work towards a distributed architecture for our network telescope that will incorporate data from disparate locations. With a distributed monitoring system observing a large fraction of the IP address space and methods of de-convolving observed data into an original attack, we will analyze biases in worm and denial-of-service attack patterns, including biases in random number generation and intentional weighting designed to increase worm propagation speed.

4. Data publication

   In conjunction with a previously funded NSF proposal, “Correlating Heterogeneous Measurement Data to Achieve System-Level Analysis of Internet Traffic Trends” we will annotate, archive, and make publicly available datasets collected for security event research. Combined with our published results, trace datasets in the repository will be available for others to confirm and extend our results. The network security and Internet measurement communities at large all have the potential to directly benefit from the datasets and analysis funded by this proposal due to this unprecedented level of access to research materials.

6.1 Milestones

Year 1: Using previously collected denial-of-service data, we plan to investigate trends in attack activity over the past two years. We will begin an in-depth study on how network telescope size and location effect what network events and properties can be accurately measured. We will develop a schedule and methodology for long-term collection of denial-of-service, worm and scanning data from our network telescopes and begin archiving this data.

Year 2: We plan to work on modifications to our network telescopes to support reacting to traffic in real-time. An unobtrusive probing test would be used for denial-of-service attack damage quantification. We will investigate trends in worm spread and research techniques for early worm detection and classification.

Year 3: We will continue to work on quantifying worms and denial-of-service attacks using reactive approaches. We will investigate trends in the efficacy of deployed security measures. We plan to develop a methodology for deployment of a distributed network telescope.

Across all three years, we will disseminate research results and make datasets and tools available to the research community as appropriate.

6.2 Management

Graduate Students: Co-PIs Stefan Savage and Geoff Voelker will guide graduate students in appropriate research directions under this proposal. Colleen Shannon, a researcher with CAIDA, will supervise student work on sensitive datasets and provide mentoring in the analysis of large (e.g. hundreds of gigabytes) network datasets.

Communication: Overall project direction will be facilitated by regular status and planning meetings among the PIs and Colleen Shannon. Graduate students will meet with their advisors/supervisor as necessary and will present intermediary results at CAIDA and CSE Systems Group research meetings.

Data Collection: Raw, unencoded trace data will be kept on CAIDA machines with limited access[2] if such data is considered sensitive to either the owner of the monitored network or is believed by the PIs to constitute a risk to others (e.g. provide a list of machines vulnerable to a particular exploit). Where possible, sanitized or anonymized data will be used. Due to their experience and trust by the community, CAIDA staff will manage the collection, storage and anonymization of data. This project will make use of already funded data repositories under CAIDA's "Correlating Heterogeneous Measurement Data to Achieve System-Level Analysis of Internet Traffic Trends" project to store datasets. Note that during August 2001, collecting only packet header data for Code-Red probes to our network telescope resulted in 0.5GB of compressed raw data per hour.

Dissemination of Results: In addition to publication of research results through scientific conferences journals, we will make results of this project available in several other ways. As appropriate results will be presented at operational networking meetings (e.g. NANOG). Tools and datasets developed during the course of this project will be made available via the CAIDA web site.

7 Conclusion

We believe that the successful completion of this effort will produce fundamental insights into the nature of malicious behavior on the Internet and consequently the best directions for mitigating that behavior. In less than three years, large-scale Internet attacks such as denial-of-service flooding and self-propagating worms have emerged as critical threats to our communications infrastructure. Moreover, during this same period these attacks have undergone rapid evolution and refinement. We can no longer afford to analyze each new attack innovation post facto with microscope and tweezers. It has become essential for the Internet community to develop meaningful and up-to-date quantitative characterizations of attack activity such as those that we have proposed.

Our initial ideas, presented to the security community at the 2001 USENIX Security conference (where we received the best paper award) and to the networking community at the 2002 ACM SIGCOMM Measurement workshop and the 2003 IEEE INFOCOM conference, have been warmly received and widely cited. We hope this grant will enable us to further generalize our approach, refine our analysis techniques, and expand our measurement infrastructure to the point where our results can become a decision making resource for researchers, security professionals, and policy makers.

[2] Additionally, all users with access to the machines must sign and abide by usage agreements including an understanding of the University of California's Electronic Communications Policy. See "Computer Facilities Usage Agreement", "Data Privacy Agreement" and "Privileged Access Usage Agreement" at http://www.caida.org/home/legal/.

References

[1] B. Huffaker, D. Plummer, D. Moore, and k. claffy, “Topology discovery by active probing,” in *Symposium on Applications and the Internet (SAINT)*, (Nara, Japan), SAINT, Jan 2002. `http://www.caida.org/outreach/papers/2002/SkitterOverview/`.

[2] B. Huffaker, M. Fomenkov, D. Moore, E. Nemeth, and k. claffy, “Measurements of the Internet topology in the Asia-Pacific Region,” in *INET ’00*, (Yokohama, Japan), The Internet Society, 18-21 July 2000. `http://www.caida.org/outreach/papers/2000/asia_paper/`.

[3] B. Huffaker, M. Fomenkov, D. Moore, and k. claffy, “Macroscopic analyses of the infrastructure: measurement and visualization of Internet connectivity and performance,” in *PAM 2001*, (Amsterdam, Netherlands), RIPE NCC, Apr 2001. `http://www.caida.org/outreach/papers/2001/SkitViz/`.

[4] B. Huffaker, M. Fomenkov, D. Plummer, D. Moore, and k. claffy, “Distance Metrics in the Internet,” in *IEEE International Telecommunications Symposium (ITS)*, (Brazil), IEEE, Sept 2002. `http://www.caida.org/outreach/papers/2002/Distance/`.

[5] M. Fomenkov, k. claffy, B. Huffaker, and D. Moore, “Macroscopic Internet Topology and Performance Measurements From the DNS Root Name Servers,” in *Usenix LISA*, (San Diego, CA), Usenix, 4-7 Dec 2001. `http://www.caida.org/outreach/papers/2001/Rssac2001a/`.

[6] C. Dovrolis, P. Ramanathan, and D. Moore, “What do packet dispersion techniques measure?,” in *INFOCOM 2001*, (Alaska), Apr 2001. `http://www.caida.org/outreach/papers/2001/consti/`.

[7] C. Shannon, D. Moore, and k. claffy, “Characteristics of fragmented IP traffic on Internet links,” in *ACM Internet Measurement Workshop 2001*, (San Francisco, CA), Nov 2001. `http://www.caida.org/outreach/papers/2001/Frag/`.

[8] C. Shannon, D. Moore, and k. claffy, “Beyond Folklore: Observations on Fragmented Traffic,” *To appear in IEEE/ACM Transactions on Networking*, Dec 2002. `http://www.caida.org/outreach/papers/2002/Frag/`.

[9] D. Moore, K. Keys, R. Koga, E. Lagache, and k. claffy, “CoralReef software suite as a tool for system and network administrators,” in *Usenix LISA*, (San Diego, CA), Usenix, 4-7 Dec 2001. `http://www.caida.org/outreach/papers/2001/CoralApps/`.

[10] K. Keys, D. Moore, R. Koga, E. Lagache, M. Tesch, and k. claffy, “The architecture of CoralReef: an Internet traffic monitoring software suite,” in *PAM 2001*, (Amsterdam, Netherlands), RIPE NCC, Apr 2001. `http://www.caida.org/outreach/papers/2001/CoralArch/`.

[11] D. Moore, R. Periakaruppan, J. Donohoe, and k. claffy, “Where in the world is netgeo.caida.org?,” in *INET ’00*, (Yokohama, Japan), The Internet Society, 18-21 Jul 2000. `http://www.caida.org/outreach/papers/2000/inet_netgeo/`.

[12] D. Moore, G. Voelker, and S. Savage, “Inferring Internet Denial-of-Service Activity,” in *Usenix Security Symposium*, (Washington, D.C.), Aug 2001. **[Best Paper Award]**. `http://www.caida.org/outreach/papers/2001/BackScatter/`.

[13] D. Moore, C. Shannon, G. Voelker, and S. Savage, “Internet Quarantine: Requirements for Containing Self-Propagating Code,” in *In submission*.

[14] D. Moore, C. Shannon, and J. Brown, “Code-Red: a case study on the spread and victims of an Internet worm,” in *ACM Internet Measurement Workshop 2002*, (Marseille, France), Nov 2002.

[15] D. Moore and C. Shannon, “The spread of the code-red worm (crv2).” `http://www.caida.org/analysis/security/code-red/coderedv2_analysis.xml`.

[16] D. Moore, G. M. Voelker, and S. Savage, “Inferring Internet Denial-of-Service Activity,” *Usenix Security Symposium*, 2001.

[17] S. Staniford, V. Paxson, and N. Weaver, “How to 0wn the Internet in Your Spare Time,” in *Usenix Security Symposium*, Aug. 2002.

[18] Computer Emergency Response Team, “CERT Advisory CA-1996-21 TCP SYN Flooding Attacks.” `http://www.cert.org/advisories/CA-1996-21.html`, Sept. 1996.

[19] V. Paxson. Personal Communication, Jan. 2001.

[20] D. Moore, “Network telescopes: Observing small or distant security events,” Aug. 2002.

[21] T. Darmohray and R. Oliver, “Hot Spares For DoS Attacks,” *;login:*, vol. 25, July 2000.

[22] eEye Digital Security, “Advisories and Alerts: AD20010618.” `http://www.eeye.com/html/Research/Advisories/AD20010618.html`.

[23] Microsoft, “A Very Real and Present Threat to the Internet.” http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/topics/codealrt.asp.

[24] eEye Digital Security, “Advisories and Alerts: .ida “Code Red” Worm,” July 2001. `http://www.eeye.com/html/Research/Advisories/AL20010717.html`.

[25] Silicon Defense, “Code Red Analysis page.” `http://www.silicondefense.com/cr/`.

[26] H. Project, “Know your enemy: Honeynets,” Nov. 2002.

[27] H. W. Hethcote *SIAM Review*, vol. 42, no. 4, pp. 599–653, 2000.

[28] J. O. Kephart and S. R. White, “Directed-graph epidemiological models of computer viruses,” in *IEEE Symposium on Security and Privacy*, pp. 343–361, 1991.

[29] David Meyer, “University of oregon route views project.” http://www.routeviews.org/.