The Internet Motion Sensor: A distributed global scoped Internet threat monitoring system

Evan Cooke, Michael Bailey, David Watson, and Farnam Jahanian
Electrical Engineering and Computer Science Department
University of Michigan
{emcooke, mibailey, dwatson, farnam}@eecs.umich.edu

Jose Nazario
Arbor Networks
jose@arbor.net

Abstract
Networks are increasingly subjected to a broad spectrum of threats that impact the reliability and availability of critical infrastructure. In response, researchers and network operators have increasingly relied on monitoring to characterize and track these threats. This paper introduces the Internet Motion Sensor (IMS), a globally scoped Internet threat monitoring system whose goal is to measure, characterize, and track threats. The dark address sensors in the IMS extend simple passive capture using a novel transport layer service emulation technique to elicit payloads across all services, thereby addressing the issue depth of service coverage. To achieve breadth of coverage, the IMS employs a distributed infrastructure and utilizes sensors that are aware of their address diversity and their position in the actively routed topology. Finally, the IMS uses an innovative signature encoding and data warehousing system combined with a hierarchical architecture to realize a system that is not only time and space efficient, but is also scalable to a global deployment. We explore the various architectural tradeoffs in the context of a 3 year deployment across multiple dark address blocks ranging in size from /24s to a /8. We show how the current architecture emulates services across a diverse set of routed and address topologies in a scalable manner. Results from three recent events are presented to illustrate the utility of such a system: the SCO Denial of Service attacks (December, 2003), the Blaster worm (August, 2003), and the Bagle backdoor scanning efforts (March, 2004).

Keywords: network security, blackhole monitoring, globally scoped threats, zero-day worms, service emulation, DDoS

1. Introduction

Networks are increasingly subjected to a broad spectrum of threats that impact the reliability and availability of critical infrastructure. These threats include distributed denial of service attacks, fast moving worms, and routing exploits. First and foremost they are globally scoped, respecting no geographic or topological boundaries. Complicating matters, they are sometimes zero-day threats, exploiting vulnerabilities for which no signature or patch has been developed, making detection and mitigation of these threats problematic. Third, these threats are evolutionary, with each worm or attack learning from previous failures, spawning an arms race between the network defenders and the attackers. Finally, many of these threats are exceptionally virulent, propagating to the entire vulnerable population in the Internet in a matter of minutes, rendering human response impractical. Researchers are attempting to address these threats by investigating new methods for monitoring and analysis.

This paper introduces the Internet Motion Sensor (IMS), a globally scoped Internet threat monitoring system. The goal of the IMS is to measure, characterize, and track threats on the global Internet. The IMS is composed of topology-aware dark IP network sensors [21] and aggregators. Each sensor monitors a block of dark address space that, while routable, contains no active hosts. Because each sensor only monitors traffic that has no operational value, any data collected is the result of outside processes like Internet worms, backscatter traffic, or scans. These blocks of space can vary from an individual host to wide address measurement such as a /8 network which contains millions of contiguous addresses. Like BGP off-ramping techniques [25, 9] this system leverages the existing network infrastructure to provide a wealth of pre-filtered data for analysis. The IMS is comprised of many of these sensors deployed in key locations to observe and characterize security threats on a global scale. The data provided by these sensors allows the IMS to quantify the prevalence, virulence, and persistence of new and ongoing threats.

We believe there are three key problems that must be addressed when building an Internet threat monitoring system. The first issue is service coverage. There are a myriad of services on the Internet today, and a monitoring system must choose what range of services to emulate and with what fidelity to emulate them. An ideal system would reproduce all current and future services with exactly the same behavior as all possible end-hosts. Such a system is impractical

because of resource constraints so there must be tradeoff. Without using live hosts, the challenge becomes what services to emulate and how to emulate them. The second major issue is topological diversity. If the goal of a monitoring system is global reach, then the sensors should be placed to achieve the greatest possible view of threat activity. In particular, worms and other automated malware typically propagate along address ranges so address diversity is important. Moreover, threats that use the Internet as a propagation medium encounter a dynamic routing topology. Sensors should be aware of this context and be positioned in such a fashion as to maximize coverage of Internet topology. The third major issue faced by a globally scoped threat monitoring system is scalability. A system designed to monitor threats on global scale must not only be able to monitor large blocks of contiguous address space, but also efficiently aggregate data from hundreds of those sensors.

The IMS is designed to address these problems and provide a scalable and extensible architecture for Internet threat monitoring. Firstly, in order get the widest possible service coverage, each sensor passively collects all UDP and ICMP traffic and uses a lightweight responder to elicit the initial packet of each TCP connection. This approach effectively emulates the establishment of TCP transactions providing the maximum service coverage without the need for maintenance-heavy modules. Second, the distributed nature of the IMS architecture allows us to explore the benefits of deployment in both address and topologically diverse locations. Finally, the IMS is very much decentralized making it extremely scalable and extensible. This is accomplished through a query engine that effectively turns the sensor network into a large distributed database. Queries are submitted to an aggregator that acts as a proxy to send out requests and collect and forward replies. A benefit of this design is the ability for operators to see a real-time view of their network and potentially hundreds of other networks around the world. Furthermore, since the aggregators do not actually store any data, any organization can operate an aggregator meaning there is no central point of failure.

This paper makes three unique contributions. The first is a characterization of the tradeoffs inherent in building a globally scoped monitoring system. Secondly, this insight is used to motivate a new architecture to achieve global visibility and differentiate and identify new threats. Finally, utility of this approach is demonstrated using a series of analyses based on data collected from a multi-year distributed deployment.

The paper is organized as follows: Section 2 describes the field of threat monitoring and introduces a framework for discussing design tradeoffs. Section 3 details the architecture and core components of the IMS. Section 4 describes the initial IMS deployment and shows some preliminary analysis. Finally, Section 5 discusses future directions and several interesting extensions to the signature generation technology.

2. Background and Related Work

With so many threats to global Internet security, characterizing, monitoring, and tracking these threats is quickly becoming critical to the smooth running of individual organizations and the Internet as a whole. Traditionally, approaches to threat monitoring fall into two broad categories, host based monitoring and network based monitoring.

Host based techniques fall into two basic approaches, forensics and host based honeypots. Antivirus software [5] and Host Based intrusion detection systems [7] seek to alert users of malicious code execution on the target machine by watching for patterns of behavior or signatures of known attacks. Host based honeypots [3, 6] track threats by providing an exploitable resource and monitoring it for abnormal behavior. A major goal of honeypots [22] is to provide insight into motivation and techniques behind these threats.

The second monitoring approach is to monitor threats from the network perspective. Passive network techniques are characterized by the fact that they do little to intrude on the existing operation of the network. By far the most common technique is the passive measurement of live networks. They fall into three main categories: data from security or policy enforcement devices, data from traffic characterization mechanisms, and direct sensing or sniffing infrastructure. By either watching firewall logs, looking for policy violations, or by aggregating IDS alerts across multiple enterprises [19, 28], one can infer information regarding a worm's spread. Other policy enforcement mechanisms, such as router ACLs provide course-grained information about blocked packets. Instead of dropping these packets, CenterTrack [25] leveraged the existing routing infrastructure to collect denial of service traffic for analysis. Data collection techniques from traffic planning tools offer another rich area of pre-existing network instrumentation useful in characterizing threats. Course-grained interface counters and more fine-grained flow analysis tools such as NetFlow [4] offer another readily available source of information.

Another interesting approach to passively collecting data comes from traffic to unused (or dark) address space. Because this space has no legitimate hosts, traffic destined to the space there is the result of malicious activity or misconfiguration. The most common application of this technique is the global announcement and routing of unused space to a collection infrastructure that records the incoming packets [14, 16, 21].

The second networked monitoring approach uses active network perturbation to determine the scope and propagation of threats. This is typically done to elicit a behavior that is only visible by participating in a network or application session. Projects like honeynet [23] and iSink [29], and software like honeyd [18] are used to bring up networks of honeypots; places designed to capture information about intrusions in a controlled environments.

2.1 Breadth, Depth, and Cost

The techniques described above provide varying amounts of intelligence regarding a threat. Some systems capture of all the events involved in a particular incident while others record only a connection attempt. Some systems only have visibility into local events, while other are capable of monitoring globally scoped threats. These tradeoffs, which we refer to as depth and breadth, are bounded by their associated costs.

Breadth refers to the ability of the system to detect threats across hosts and across operational and geographic boundaries. At one extreme of the breadth axis is the threat view of a single host while at the other is the view of all network-based threats on a global scale. One early approach to achieving globally scoped visibility was the measurement of wide address blocks [14, 21, 29]. This technique is attractive in that it can easily view a large percentage of the total IPv4 address space and has been effective at characterizing new threats [15, 20]. However, given the finite size of the IPv4 address space, it is important to explore new methods of obtaining breadth. In this paper we explore two advantages of sensor placement based on address and topological diversity.

Depth defines the extent to which a sensor emulates the services and characteristic of a real host, similar to the interaction spectrum in honeypots [22]. At one extreme of the depth axis is an instrumented live host while at the other is the completely passive capture of packets. Multiple points in this spectrum can be utilized simultaneously, as shown by Barford *et al*. [29]. In this paper, however, we demonstrate an extension to passive techniques [14] that gains additional intelligence without emulating services to the extent of previous active methods [23, 18].

Figure 1 shows the range of breadth and depth and introduces the concept of cost, which is proportional to the product of breath and depth for any particular approach. Cost is qualitative metric that is a consequence of any approach. This includes the cost of construction (development, deployment, maintenance), the impact on the network as a consequence of active response, and the computational and space requirements. A natural consequence of this analysis is that any system has an inherent trade-off of depth and breadth versus cost.

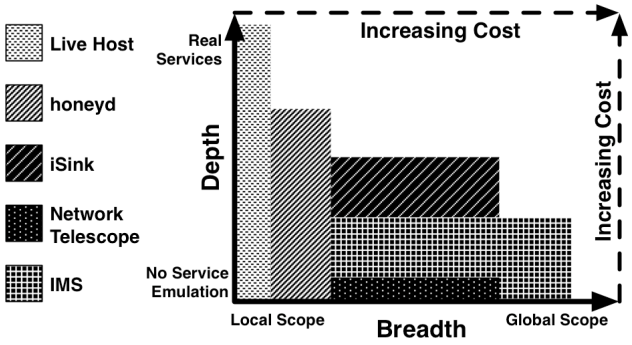


Figure 1. Breath vs. Depth classification of various Internet threat monitoring architectures

3. Internet Motion Sensor Architecture

The Internet Motion Sensor is designed to maximize breadth and depth while minimizing cost in order to facilitate a global deployment. The IMS architecture introduces a novel data point in the spectrum of Figure 1, which directly follows from the design goals for the project:

- Maintain a level of interactivity that can differentiate traffic on the same service.
- A characterization of emerging threats.
- Visibility into the Internet beyond geographical and operational boundaries.

The IMS architecture consists a set of heterogeneous sensors and one or more data aggregators as depicted in Figure 2. The sensors deployed in the IMS can be divided into two basic categories: blackhole or dark IP sensors and topology sensors. The blackhole sensors form the core of the IMS by collecting threat data and the topology sensors provide context for that information.

Each blackhole sensor monitors a dedicated range of unused IP address space. The blackhole sensors in the IMS have an active and passive component. The passive component records packets sent to the sensor's address space and the active component responds to specific packets to illicit more data from the source.

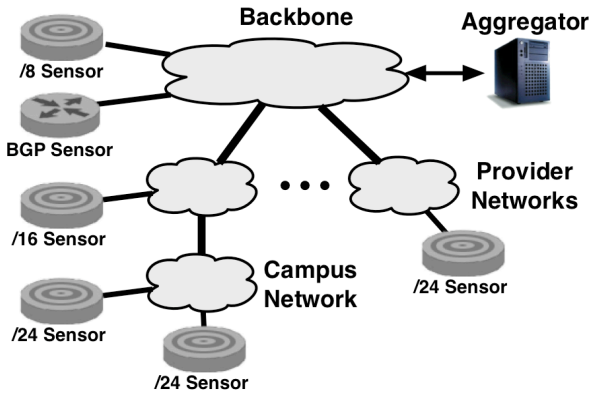


Figure 2. Internet Motion Sensor Architecture

The active component in each blackhole sensor only responds to TCP connection requests. UDP and ICMP packets do not need an active response because the initial packet contains all the information available without taking on the personality of specific services. For example, the Witty [20] and Slammer [15] worms were based on UDP in which the entire worm payload was transmitted in the first packet. TCP, on the other hand, is a connection based protocol and requires an active response to elicit payload data. A TCP session is initiated by sending a SYN packet to the destination host. The receiving host then sends a SYN-ACK packet back to the sender to begin the session. A self-propagating worm using TCP must send a SYN and receive a SYN-ACK before it can send the first packet of data and the subsequent exploit. This feature allows the IMS to capture the payload of TCP worms like Blaster [2] and Sasser [12]. In order for an individual blackhole sensor to monitor wide address spaces, the active responder component supports sampling.

The passive component of the IMS blackhole sensors is an integral part of the synthesis of data. One of the main design goals for the IMS was to support real-time trending and analysis necessitating a novel approach to data processing. The problem is that raw traces gathered on each sensor can approach a gigabyte of data per day. Attempting to transmit and store that data for a large number of sensors while supporting real-time data analysis requires very significant infrastructure. The obvious solution is to distribute the workload. Each blackhole sensor in the IMS is responsible for gathering and archiving data, performing queries on its local data store, and generating alerts that are sent to the aggregator.

All IMS sensors gather data in a format to facilitate fast queries using relational database techniques used in a data warehousing. Each row in a fact table stores basic information about the connection like *<srcip, srcport, dstip, dstport,*

protocol> and also pointers to other tables containing the full headers and the full payload. By separating connection information from full header and payload data, common queries on connection information can be processed quicker.

Storing the full payload for every packet received has significant space requirement, so the IMS uses a novel payload storage approach. When a blackhole sensor receives a packet with a payload it first computes the MD5 checksum of the payload (without network headers) and compares it against the checksum of all the other packets it has seen in the past day. If the checksum, or signature, has already been recorded, the passive capture component logs the signature but does not store the payload. If the signature is new, the payload is stored and the signature is added to the database of signatures seen in that day. The signature database uses an array-based suffix tree with an alphabet of 256 characters, so insertions and searches are performed in constant time. Since the signature database is flushed and zeroed once a day, it can remain in RAM making searches very fast.

The blackhole sensors are closely integrated with the routing sensors that collect topology meta-data. The routing sensors [11] are responsible for providing an accurate picture of the local routing topology. They are built around a customized daemon that is configured as an iBGP peer of the critical routers near the blackhole sensors. The routing sensors store data in standard MRT format, including both individual updates from each peer as well as occasional routing table dumps for ease of search and indexing. These sensors enable both real time views of the routing topology from each sensors as well as accurate post event analysis of data.

The next three subsections evaluate the design decisions taken in building the IMS in the context of a multi-year deployment. The three metrics used to assess the IMS in a manner comparable with existing systems are Depth/Service Coverage, Breadth/Topological Diversity, and Cost/Scalability.

3.1 Depth/Service Coverage

The first issue is depth. While existing monitoring techniques have either been completely passive or attempted full session reconstruction responding to TCP SYN's with SYN-ACK's provides the data necessary to capture a large portion of threat activity. Take for example the Blaster worm [2]. The infection and transmission method used by Blaster is relatively complicated compared to a single-packet worm like Slammer [15]. The Blaster worm first opens a TCP connection to port 135 and sends an RPC bind request. Next, an RPC request message is sent containing a buffer overflow and code to open a backdoor port on TCP port 4444. The newly infected host then sends a message via the new backdoor to download the worm payload and execute it. A condensed diagram of the transactions involved in a Blaster infection is illustrated in Figure 3a.

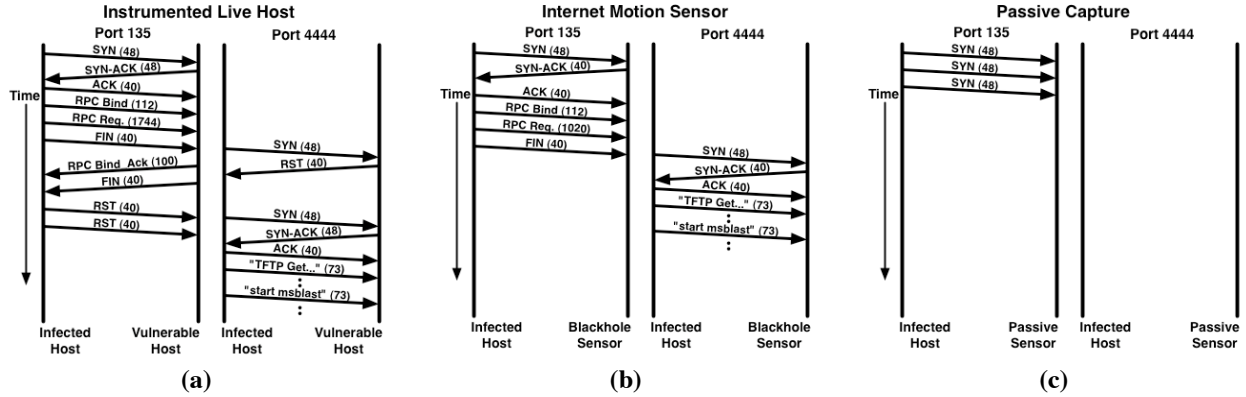


Figure 3. Blaster infection attempt captured using three monitoring techniques

Figure 3b depicts the transactions of the Blaster worm as captured by a blackhole sensor in the IMS. Observe how a single SYN-ACK on port 135 elicits not only the exploit, but also a SYN on the backdoor port. Since the IMS blackhole sensors respond to SYN packets on all ports, the worm connects to port 4444 and sends the download and start commands assuming the attack was successful. Thus, the IMS has the depth necessary to see and capture all major transactions for the Blaster worm.

Now compare the data captured by the IMS to the data recorded by a passive blackhole monitor, shown in Figure 6c. While a passive monitor might catch a single packet UDP worm like Sapphire, it will only see SYN traffic from TCP worms like Blaster. The Blaster example shows how the IMS can effectively capture session payloads for a complex worm running on a common service.

Another important illustration of the value in having more information is the extraction and classification of a new threat on a highly trafficked service. The Sasser [12] worm utilized TCP port 445, which is a heavily used service for many existing threats. Because the IMS was able to obtain and classify the Sasser payload, it was able to identify the traffic specific to this new worm as shown in Figure 5. Figure 5a shows the traffic captured on port 445 over the period of 7 days. Figure 5b shows only the traffic of the signature associated with Sasser over that same time period. Thus, the IMS is able to identify the presence of a new worm even in an extremely noisy service using payload signatures.

Another advantage of using a service agnostic approach is insight into less popular services. One example is a management application that may not be widely deployed. The population deploying this service might only be several thousand hosts on the global Internet and the obscurity of the service may mean it is unmonitored. Another example of less well know services are backdoor ports on existing worms and viruses. New threats that exploit these obscure services can avoid detection by existing network monitoring tools because they do not have the appropriate service modules. In contrast, the IMS can detect and gather significant information on this kind of threat. The IMS has been tracking an increasing traffic on TCP ports which do not have well know services as soon in Figure 4.

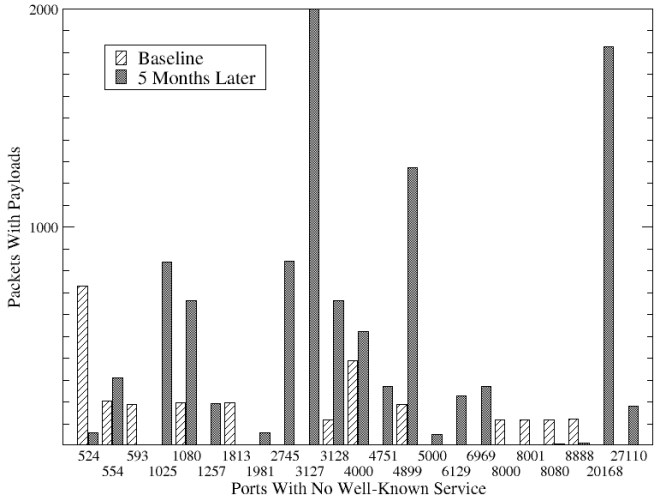
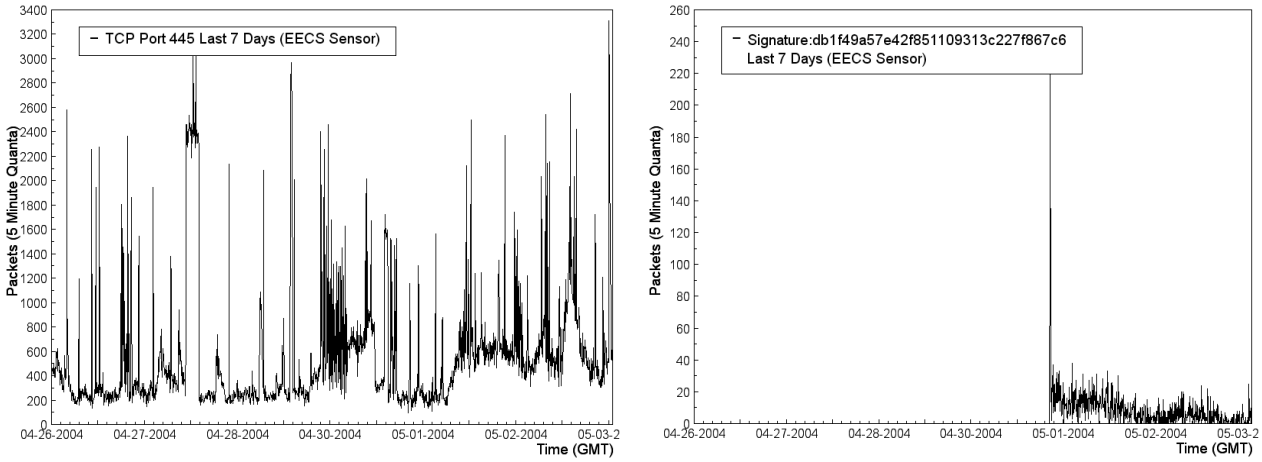


Figure 4. Increasing activity on TCP ports without well-known services

By design, the blackhole sensors that form the core of the IMS monitor an extremely wide array of services and have the depth necessary to capture significant data on these services. In particular, the IMS has the depth to capture threats against existing services like Blaster and also track obscure services like worm backdoors. Moreover, because the IMS never assumes a specific OS or application personality, there is never the problem of having to emulating services correctly and with fidelity equal to that of a real end-host.

3.2 Breadth/Topological Diversity



(a) **(b)**

Figure 5. The Sasser worm as recorded by an IMS /24 blackhole sensor

The second major architectural feature of the IMS is its position in the breadth spectrum. In designing the IMS the authors sought explore breadth by looking at topological and address diversity through sensor placement. The IMS approaches this problem by positioning sensors to capture threat traffic in key topological positions and then aggregate the data to provide a global perspective.

In order to illustrate value of address diverse deployment, consider the following example. Figure 7 shows traffic originating from a single /16 network from the perspective of three different /24 blackhole sensors. One /24 sensor (Figure 7a) is located in the same /16 as the traffic source. The second /24 sensor (Figure 7b) is located within the same /8 as the traffic source. Finally, the final sensor (Figure 7c) is located in completely separate /8 from the traffic source. This topology is depicted in Figure 7d. The fascinating implication of this graph is that placing sensors near the source of traffic can yield substantially different results than those placed further away.

An interesting artifact of this is that a single /8, such as that of the original IMS deployment, is disadvantaged in that it cannot see local traffic if the entire space is unused. Local scanning preferences and other worm characteristics are lost if monitored address blocks are not near live (infected) hosts. The large size of the /8 does, however, enable analysis not available for many dispersed /24 sensors. For example, the /8 in IMS observed a series of scans which only scanned a single host within any /16 network. It turned out be result of a broken scanning tool but the point is that kind of data would be difficult to extract from a network of smaller sensors.

In addition to address diversity, it is also important to consider routed topology. The IMS explicitly includes routing sensors in the architecture. This enables the system to compensate for properties of routed topology that make it difficult to infer globally scoped events from a limited number of fixed sensors. The first of these, spatial diversity refers to the fact that AS and router level topologies are different when view from different parts of the network [1]. Secondly, temporal dynamics refers to the notion that the active routed topology varies over both the short and long term [11].

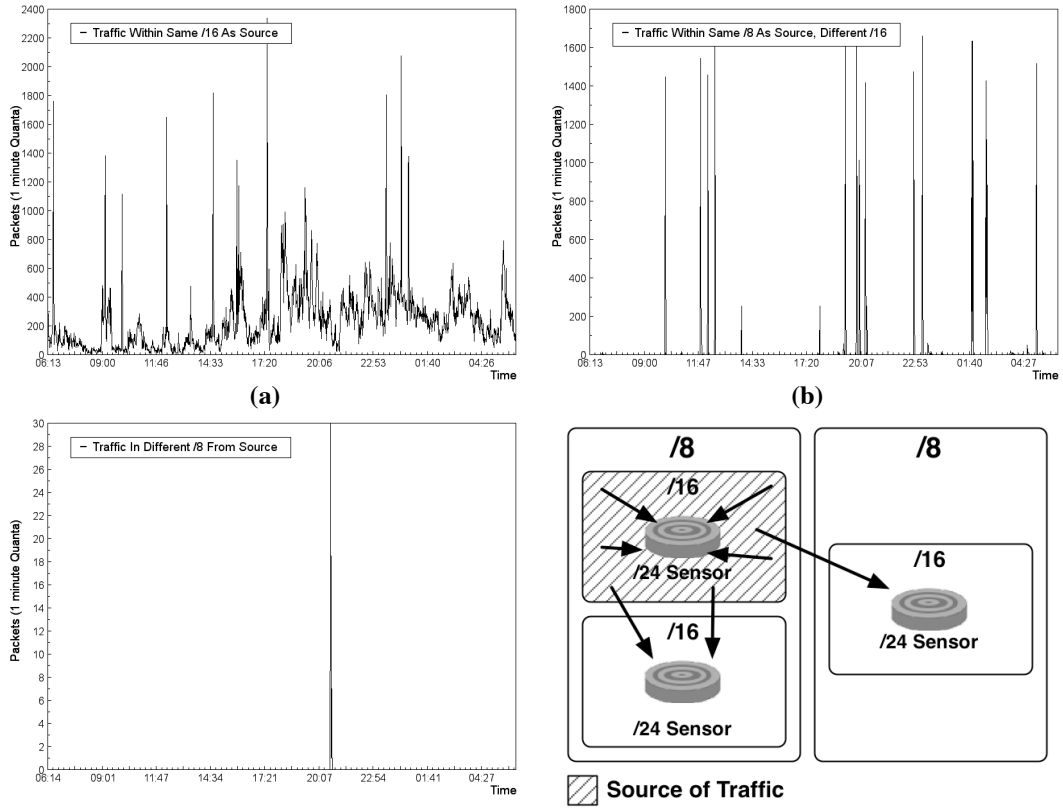


Figure 7. Traffic from a single /16 capture from 3 different perspectives

While understanding these properties of routed topology are interesting research areas, a failure to incorporate them into the analysis of inferred global events can be problematic. These pitfalls fall into a small number of categories:

- **Drift.** Post analysis and forensics that rely on topology will be skewed if historic topology information is no longer available.
- **Availability**. Fine-grained topological awareness may be necessary to understand those threats that may be correlated with or causally related to infrastructure impacting events.
- **Policy**. Monitoring at a specific location may result in too great a dependence on a single group of upstream service providers. These providers may apply (different) policies to deal with a specific threat.
- **Topology**. Distances from target populations, address proximity, lack of hosts, all may impact the propagation of the worm and may miss hit-list worms

Figure 6 shows the growth and decay of the Witty worm as viewed by a subset of the IMS system. Even though the 3 sensors are address diverse (deployed on 3 separate /16s, and 2 separate /8s) we see that they all suffered a loss of data for a twenty-minute period. Investigation revealed this drop was the result of a policy decision by a shared upstream service provider.

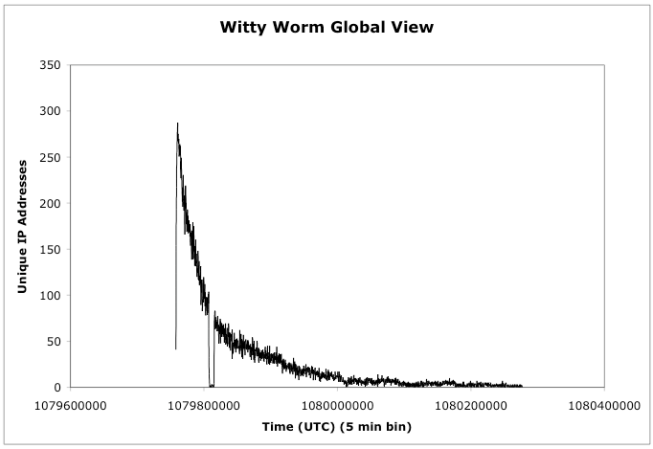


Figure 6. The growth and decay of the witty worm as seen by subset of IMS

Thus, when considering the deployment of blackhole sensors it is important to consider placement within the Internet address space and also to be aware of the routed topology. In particular, one must be aware of the bias toward local traffic in active network. In addition, one must also be concerned with placement in the routed topology. Reliance on a single provider offers a very restricted view and can sometimes lead to unplanned outages.

3.3 Cost/Scalability

The final major architectural feature of the IMS is scalability. Individual sensors are designed to be efficient and scale to wide address blocks. The aggregator hierarchy also provides scalability in the number of sensors that may be deployed. At the sensor level, the blackhole sensors use payload signatures so only one copy of each payload is ever stored on a per-day basis. When a blackhole sensor receives a packet with a payload it first computes the MD5 checksum of the payload (without network headers) and compares it against the checksum of all the other packets it has seen in the past day. This approach offers a factor of two savings in disk (Figure 8) and the hit rate on the signature cache typically tops 96% (Figure 9). The implication is that a large number of the payloads seen each day are exactly the same.

The IMS was also intended to scale efficiently as additional sensors are added to the system. In its current form, the IMS runs in almost constant time regardless of the number of sensors. Alerts are extremely lightweight, and a single aggregator can handle alerts from thousands of sensor. Data queries would cause a centralized version of the system to become practically unfeasible with over ten sensor but the queries in the IMS are delegated to the sensors themselves. Thus, the time for a query across the entire system is limited by the time of the slowest sensor to respond. In addition, the distributed nature of the system makes it extremely reliable. Sensors are not tied to specific aggregators, so if a sensor or aggregator fails the rest of the system operates normally. During periods of severe routing instability this can be critical as operators can utilize as much of the system as is reachable.

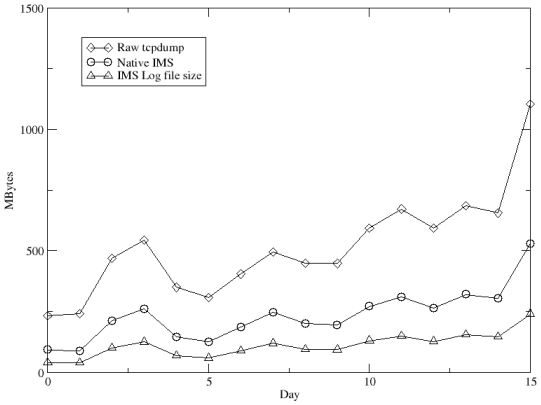


Figure 8. IMS and tcpdump log file sizes over a 16 day period.

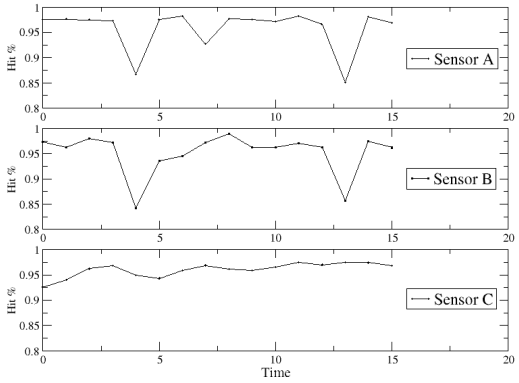


Figure 9. Signature database hit rate over 16 day period on three blackhole sensors. Fluctuations in the hit rate in two of the sensors correspond with an attack on a MySQL Server service.

4. Deployment Observations and Analysis

The IMS was deployed in 2001 as part of a global monitoring system developed by researchers at Arbor Networks and the University of Michigan. This initial installation was built to monitor an unused /8 network, or approximately 1/256th of the Internet address space. Between 2001 and 2003 this system processed several hundred petabytes of network data including recording millions of scan and backscatter events. It was able to characterize numerous Internet worms such as CodeRed, Nimda, Sapphire, and Blaster. In 2003 this system was expanded to embody the architecture presented in this paper. At the time of writing the IMS consists of 8 monitored network blocks including one /8, two /16s, and five smaller sized blocks. Commitment has been received for an additional 30 deployments over the rest of this year.

In this section we show how the IMS was used to investigate several critical threats to the security of the Internet. We examine three distinct events that represent a breadth of security issues and highlight the unique architecture characteristics of IMS.

4.1 Internet Worms

Internet worms represent a class of security threats that seek to execute code on a target machine by exploiting vulnerabilities in operating system or application software [17]. Unlike viruses, however, worms do not rely on attaching themselves to files to propagate. Rather, these worms stand-alone and propagate by using the network to scan for other potentially vulnerable hosts and exploit them without user interaction.

Globally scoped network monitoring systems, such as the IMS, are necessary to characterize, measure and track these threats. The IMS has been able to provide valuable insight into a variety of worm behaviors, including:

- **Worm Virulence**. How much traffic resulted from this worm? What routers or paths were most congested by this worm?
- **Worm Demographics**. How many hosts were infected? Where are these hosts geographically, topologically, and organizationally? What operating system are the infected hosts running? What is their available bandwidth?
- **Worm Propagation**. How does the worm select its next target?
- **Community response**. How quickly was policy employed? Which organizations were affected quickest and who responded quickest? Who is still infected?

As an example of these types of insights, consider the following brief analysis of the Blaster worm. The Blaster worm affected Windows 2000 and XP systems running DCOM RPC services and used a publicized buffer overflow vulnerability to run arbitrary code on the target machine. The worm would sequentially scan 255 contiguous addresses for services listening on TCP port 135. 60% of the /24 networks to scan were randomly generated and 40% of the

networks were located within the same host /16 network as the affected system. The IMS was able to measure the release and propagation of the Blaster worm as it attempted to affect random hosts.

The 7-day period of observations surrounding the release of the Blaster work indicates a clear 3-phased worm cycle. The first part is the growth phase of the worm in which the number of TCP port 135 scans increased from a baseline rate to hundreds of thousands per hour. The second phase of this set of observations is the decay in the number of observed TCP port 135 scans as large-scale filtering was implemented to halt the worm's spread. The third phase of the worm activity is the persistence of the Blaster worm after its outbreak. This activity has continued through early 2004.

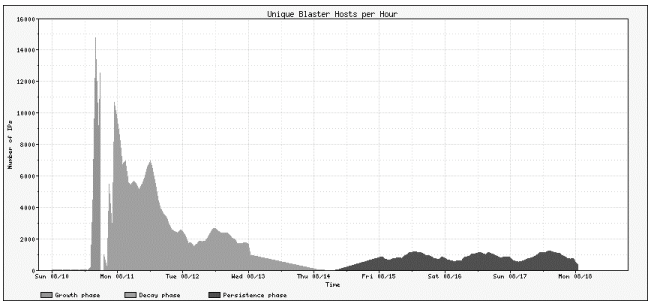


Figure 10. A snapshot of Blaster worm showing the three phases of the worm lifecycle.

In this one-week period of measurement, the IMS system observed over 286,000 unique IP addresses displaying the characteristics of Blaster activity. Inspection of the DNS top-level domains (TLD) from the reverse lookups shows that the .net and .com domains were hit most heavily, with .jp as the third more popular unique TLD. Furthermore, approximately 10% of the hosts observed were identifiable as dynamically assigned addresses.

At its highest pace, the Blaster worm was spreading with a doubling time of less than 2.3 hours. This value may be overestimated due to the truncated propagation phase of the worm. Fitting a sigmoidal population growth equation to this phase of the data shows a maximal growth rate of 40,000 hosts per hour. The second major phase of the data collection monitored the containment of the Blaster worm. Starting within 8 hours of the worm's initial outbreak, the number of unique hosts per-hour scanning for TCP port 135 began to diminish. This loss of activity fits a simple exponential decay model. The half-life of these observations is approximately 10.4 hours and continued for five days, through the end of the workweek.

Continued Blaster activity displays a circadian pattern, with peak activity occurring near 05:00 GMT. Between 1000 and 2000 hosts per hour are seen every day starting in late August, 2003, and continuing through early 2004. This pattern suggests that these sources are power-cycled every day and do not remain on continuously. An inspection of the reverse DNS entries for these hosts indicates a global source distribution.

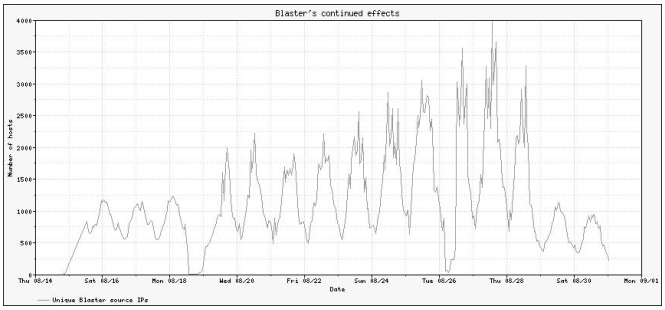


Figure 11. Unique Blaster hosts per hour immediately following the decay phase. The growth is attributed to the appearance of the Welchia worm on August 18th. The last two days represent the steady state seen for the next several months.

The Blaster worm was not as aggressive as the Slammer or Witty worms, but it did spread at a pace comparable to worms such as Code Red and Nimda. Its appearance and continued presence on the Internet shows the scope of any worm event on the Internet.

Worms are a excellent example of how the IMS provide additional insight into these threats. Port 135 traffic was observed at the IMS long before the growth of Blaster, although in much smaller magnitudes. Our unique choice of service emulation depth allowed us to elicit enough of the payload to differentiate this scan traffic from the new worm activity. In addition the IMS also has the visibility to discriminate between various worms payloads of the Blaster variants. In the case of the Blaster variants, the payloads performed different activities that required different reactions from network operators to remediate (e.g. one such payload included a denial of service attack).

4.2 Scanning

The second example of analysis enabled by this architecture focuses on scan activity. Scanning of networked computers has been around almost as long as networked computers have existed. Benign types of scans were originally used to verify network connectivity (ICMP) or forwarding paths. As applications and services began to use the network, scans for potential vulnerabilities were developed. Individuals looking for services that were vulnerable to attack scanned networks, exploited those servers, and gained control of the computing resources. With the advent of auto rooters, and complex scanning tools, probes to networks from other machine on the Internet are now a routine occurrence.

One artifact of more recent worms is that after compromising a system, these worms commonly install backdoors in the system they infect. These backdoors have long been a hypothetical source of personal data, computation and network resources. One of the more interesting applications of the IMS has been in investigating the degree to which hackers have been trying to utilize this potential resource through secondary infections.

Starting on approximately March 20, 2004, IMS began tracking significant amounts of scanning for backdoor ports left by widespread variants of the Bagel [27] and MyDoom [26] mail-based worms. The patterns of sources for this traffic show that they are widespread. The payloads identified suggest that these hosts may be undergoing opportunistic attacks.

Bagle and MyDoom are families of SMTP-based worms that began spreading in early 2004 and propagated via mass mailer routines. Both of these mail-based worms have many variants that have rapidly appeared in a short time span, with new variants appearing almost daily. The relationship between these families is interesting and reveals a fight in the virus world between authors. Some of these mail-based worms attempted to uninstall the others, disrupting their spread.

Each of these malware families listen on TCP ports as a backdoor mechanism for remote contact of the hosts. In the case of many of the Bagle variants, it is TCP port 2745. In the case of the MyDoom family of mail-based worms this port is typically TCP port 3127. Access to these ports could be used to upload arbitrary software to an affected host or execute arbitrary commands.

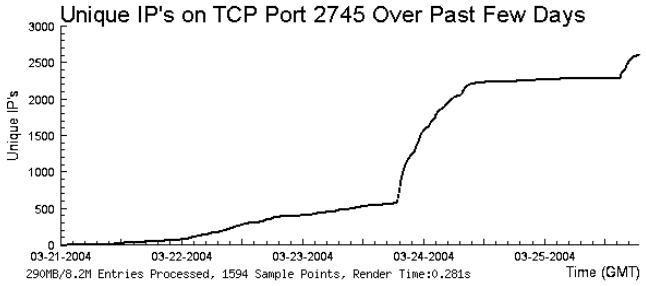


Figure 12. Propagation of 2745/TCP scanning. This figure shows the cumulative number of source IP addresses contacting TCP port 2745.

The top payload captured using IMS to port 2745/TCP (backdoor ports left by the Bagle.C-G and Bagle.J-K mail-based worms) shown in Figure 12. Scans against other ports open by the Bagle variants (including 6777 for Bagle.A, 8866 for Bagle.B, and 11117 for Bagle.L) have not been observed in any appreciable quantities.

```
43 ff ff ff 30 30 30 01 0a 28 91 a1 2b e6 60 2f
32 8f 60 15 1a 20 1a 00
```

Figure 13. Top payload against port 2745/TCP seen in scanning. This hexdump shows the most frequently observed payload of the scans against 2745/TCP.

Note that this signature in Figure 13 differs from the Bagle removal mechanism described by Joe Stewart [24]. The similarity of the first four bytes (\x43\xff\xff\xff) must be noted; after that point the signatures are divergent. This may indicate a common command or authentication byte sequence.

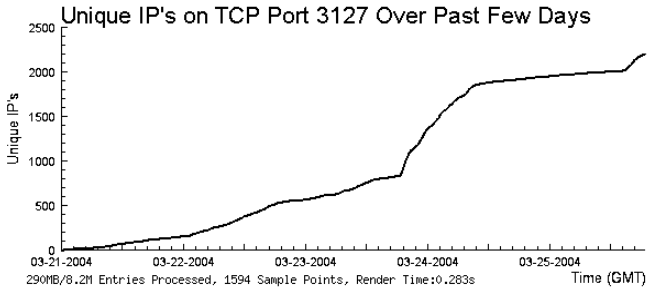


Figure 14. Propagation of 3127/TCP scanning. This figure shows the cumulative number of IP addresses contacting TCP port 3127.

The top payload for the MyDoom worm scans (target port is 3127/TCP) appears to be a UPX packed binary, suggesting new software is being uploaded. The origin and function of this binary is unknown. Both the source and destinations of these scans against TCP port 3127 appear to be widespread. Sources are in many global networks, typically in consumer broadband networks and academic networks. The IMS has not observed significant activity in this time period against additional MyDoom ports, including TCP ports 3128-3198.

One interesting artifact of both the worm infections as well as the scanning activity is it demonstrated the utility of the service coverage depth selected by the IMS. The original worm infection created novel services (the backdoors) on the target machines. Without the ability to respond as if these new services where running, the IMS would never have been able to collect the payload and analyze this activity.

4.3 Distributed Denial of Service Attacks

The final example of enabled analysis examines Denial of Service attacks. Denial of Service attacks seek to deny legitimate users access to resources. Denying service typically takes the form of either crashing a computing resource through some bug in the software implementation or by consuming a finite resource. Distributed Denial of Service attacks (DDoS) are a subset of this class of attacks that rely on a typically large number of end hosts to consume network resources (e.g. host connection queues, available link bandwidth).

On December 10, 2003, shortly after 4PM EST a long-lived denial of service attack began against a single web server address for The SCO Group (www.sco.com). Because these attacks utilized spoofed source addresses, the IMS system was able to observe some of the backscatter from the attacks. The detection system classified the attacks as 5 discrete events. Three of these events were against the web servers for The SCO Group, one was against their FTP server, and one was against their SMTP server. Due space constraints, additional analysis such as source host fingerprinting through passive techniques and TTL based distance measurements of attacking populations are not presented, but represent part of a spectrum of analysis enabled by IMS.

Start	Duration	Type of attack
Dec. 10, 16:24 EST	902 minutes	SYN flood against port 80, www.sco.com
Dec. 11, 05:49 EST	480 minutes	SYN flood against port 21, ftp.sco.com
Dec. 11, 05:51 EST	16 minutes	SYN flood against port 25, mail.ut.caldera.com
Dec. 11, 07:28 EST	27 minutes	SYN flood against port 80, www.sco.com
Dec. 12, 12:23 EST	13 minutes	SYN flood against port 80, www.sco.com

Table 1. Discrete DDoS events targeting SCO

This events illustrates that DDoS events can be long lived and made up of several smaller events and can be combine to create a larger attacks. Furthermore, this events shows that attacks using spoofed source address are still in use.

Denial of service attacks represent an interesting demonstration of the utility of the IMS and the need for address diversity. Because attacks may randomize their sources addresses over the entire Internet, smaller swaths of address space may not be able accurately determine the scope and magnitude of an attack (e.g. a /24 may only see .000005% of the backscatter, while a /8 may see .5%).

5. Discussion

The IMS provides a scalable platform for gathering information about virulence and other detailed data about existing and emerging globally scoped Internet threats. Because of the specific tradeoffs selected in the breadth, depth, and cost, the IMS approach has several limitations.

The first issue is avoiding network blacklists in order to observe all important threats. The simple single packet response mechanism in the IMS has the side effect of appearing extremely uniform to someone probing the address block. A solution adopted by other systems [18] is to assume a virtual topology and OS personalities. The current IMS implementation attempts to avoid the blacklist problem through distribution. The more monitors that are deployed, the more difficult it becomes to identify them all successfully. One idea is to continually keep the location of the sensors in a state of flux. By breaking larger address blocks into smaller blackholes and rotating which address blocks are the active responders, it would make building such blacklists much more difficult.

A second area for future IMS improvement is the active response portion of the architecture. By utilizing a very lightweight active responder for TCP, the IMS produces a wealth of additional payload information not captured by passive monitors. Use of an active responder does not come without complications. The very act of sending a SYN-ACK reveals information about the traffic source through the IP and protocol header values. If all the SYN-ACK packets from the blackhole are identical – as is in the current IMS implementation – it is possible a worm might use network fingerprinting to determine which hosts to exploit. To achieve the greatest possible depth, future active responders should mimic a large array of OS personalities.

The third area of future improvement is obtaining additional information about systems that contact the blackholes. There is a limit to the information that can be gained from the packets exchanged during a transaction with a blackhole sensor. Passive OS fingerprinting techniques [13] can be used to speculate on the system type of the sender but it's not hard to forge those characteristics. A future system could take a more aggressive approach and attempt to port scan a subset of the hosts [8] contacting the blackhole to gain additional information. In addition to OS personality, it's also interesting to have information about the bandwidth available to a set of attackers in order to build a so called "firepower graph." A future system could use this information to estimate the impact of an attack from a given set of hosts.

The final major area for IMS improvement is improved integration with host-based systems. In particular, the metadata provided by host-based systems [10] could also be integrated into automated classification systems described earlier. While we argue that the approach used in the IMS project offers sufficient depth to identify new threats and differentiate threats within any one service, increased depth may be needed to understand some threats. Host-based approaches, including [23] and [29], can yield this information. We are actively developing ways to yield this insight with the IMS while remaining lightweight and flexible.

6. Conclusion

The IMS implements a novel approach to providing a global view of Internet operational threats. We have identified the challenges in the tradeoffs between completeness of coverage, in both breadth and depth, and the associated difficulty or cost in data acquisition and analysis. In identifying the points along these spectrums, we positioned the IMS to maximize insight while constraining costs.

We achieve a high breadth of coverage by combining the approach of a wide area monitor with topologically diverse sensor placements. This provides an aggregate view of Internet activity that can compete with a wide area monitor while also providing a view across operational boundaries. Because of this, IMS data collection is resilient to collection failure due to a single upstream provider's link loss or filtering practices. This distributed architecture also provides for an inherent scalability.

The IMS architecture generates data with depth of view that is achieved through a lightweight active responder. This allows for payload insights facilitating threat discrimination even across high traffic services such as HTTP.

Finally, IMS minimizes the cost of setup and storage, which facilitates data collection and dissemination. Because of this, new forms of investigation are available which will yield unique perspectives into threats and their evolutions.

We have demonstrated the utility of this approach through insights gained into denial of service events, opportunistic scanning and compromises, and Internet worm propagation. This approach promises to yield continued results in a scalable architecture as we grow the project in the coming months.

References

[1] Suman Banerjee, Timothy G. Griffin, and Marcelo Pias. The Interdomain Connectivity of PlanetLab Nodes. *Proceedings of the Passive and Active Measurement Workshop (PAM2004)*, Antibes Juan-les-Pins, France, April 2004

[2] CERT. CERT Advisory CA-2003-20 W32/Blaster worm. http://www.cert.org/advisories/CA-2003-20.html, 2003

[3] B. Cheswick, "An Evening with Berferd in which a cracker is Lured, Endured, and Studied", USENIX proceedings, Jan 20, 1990

[4] Cisco Systems, Inc. Cisco IOS software NetFlow. `http://www.cisco.com/warp/public/732/Tech/nmp/netflow/index.shtml`, 2004.

[5] Fred Cohen. A Short Course on Computer Viruses. John Wiley & Sons; 2nd edition (April 1994)

[6] Fred Cohen. The deception toolkit (DTK). http://www.all.net/dtk/.

[7] Stephanie Forrest, Steven A. Hofmeyr, Anil Somayaji, Thomas A. Longstaff "A Sense of Self for Unix Processes". Proceedinges of the 1996 IEEE Symposium on Research in Security and Privacy

[8] Fydor. Remote OS detection via TCP/IP stack fingerprinting. `http://www.insecure.org/nmap/nmap-fingerprinting-article.txt`, October 1998.

[9] Barry Greene. Sink hole deployments: Is it time to change our BGP BCPs? IEPG Meeting, `http://www.potaroo.net/iepg/march-2003/sink.pdf`, March 2003.

[10] Samuel T. King and Peter M. Chen. Backtracking intrusions. In *Proceedings of the 2003 Symposium on Operating Systems Principles (SOSP)*, October 2003.

[11] Craig Labovitz, G. Robert Malan, and Farnam Jahanian. Internet routing instability. *IEEEACM Transactions on Networking*, 6(5): 515–528, October 1998.

[12] Microsoft Corporation. What You Should Know About the Sasser Worm and Its Variants. http://www.microsoft.com/security/incident/sasser.asp

[13] Toby Miller. Passive OS fingerprinting: Details and techniques. `http://www.sans.org/rr/special/passiveos.php`, 2001.

[14] D. Moore. Network telescopes: Observing small or distant security events. Invited presentation at the 11th Usenix Security Symposium (SEC 02), Aug 2002.

[15] David Moore, Vern Paxson, Stefan Savage, Colleen Shannon, Stuart Staniford, and Nicolas Weaver. Inside the Slammer worm. *IEEE Security and Privacy*, 1(4): 33–39, July 2003.

[16] David Moore, Geoffrey M. Voelker, and Stefan Savage. Inferring Internet denial-of-service activity. In USENIX, editor, *Proceedings of the Tenth USENIX Security Symposium*, Washington, DC, USA, August 2001.

[17] Jose Nazario. Defense and Detection Strategies against Internet Worms. Artech House, 2003.

[18] Niels Provos. Honeyd: A virtual honeypot daemon (extended abstract). In *10th DFN-CERT Workshop*, Hamburg, Germany, February 2003.

[19] SANS Institute. Internet store center. `http://isc.incidents.org`.

[20] Colleen Shannon and David Moore. The spread of the Witty Worm. http://www.caida.org/analysis/security/witty/, 2004

[21] Dug Song, Rob Malan, and Robert Stone. A snapshot of global internet worm activity. Technical report, Arbor Networks, 2001.

[22] Lance Spitzner. *Honeypots: Tracking Hackers*. Addison-Wesley, 2002.

[23] Lance Spitzner et al. The Honeynet project. `http://project.honeynet.org`.

[24] Joe Stewart. "Bagle remote uninstall". http://www.securityfocus.com/archive/1/350568/2004-01-18/2004-01-24/2

[25] Robert Stone. CenterTrack: An IP overlay network for tracking DoS floods. In USENIX, editor, *Proceedings of the Ninth USENIX Security Symposium, August 14–17, 2000, Denver, Colorado*, Berkeley, CA, USA, 2000. USENIX.
[26] Trend Micro, Inc. WORM_MYDOOM.A. http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=WORM_MYDOOM.A, 2004.
[27] Trend Micro, Inc. WORM_BAGLE.J. http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=WORM_BAGLE.J, 2004.
[28] Vinod Yegneswaran, Paul Barford, and Somesh Jha. Global intrusion detection in the DOMINO overlay system. In *Proceedings of Network and Distributed System Security Symposium (NDSS '04)*, San Diego, CA, February 2004.
[29] Vinod Yegneswaran, Paul Barford, and Dave Plonka. On the design and use of Internet sinks for network abuse monitoring. Technical Report 1497, University of Wisconsin, Computer Science Department, 2004.