The Internet Motion Sensor: A Distributed Blackhole Monitoring System

Michael Bailey,[*] Evan Cooke,[*] Farnam Jahanian,[*†] Jose Nazario,[†] David Watson[*]

[*]Electrical Engineering and Computer Science Department
University of Michigan
{*mibailey, emcooke, farnam, dwatson*}*@umich.edu*

[†]Arbor Networks
jose@arbor.net

Abstract

As national infrastructure becomes intertwined with emerging global data networks, the stability and integrity of the two have become synonymous. This connection, while necessary, leaves network assets vulnerable to the rapidly moving threats of today's Internet, including fast moving worms, distributed denial of service attacks, and routing exploits. This paper introduces the Internet Motion Sensor (IMS), a globally scoped Internet monitoring system whose goal is to measure, characterize, and track threats. The IMS architecture is based on three novel components. First, a Distributed Monitoring Infrastructure *increases visibility into global threats. Second, a* Lightweight Active Responder *provides enough interactivity that traffic on the same service can be differentiated independent of application semantics. Third, a* Payload Signatures and Caching *mechanism avoids recording duplicated payloads, reducing overhead and assisting in identifying new and unique payloads. We explore the architectural tradeoffs of this system in the context of a 3 year deployment across multiple dark address blocks ranging in size from /24s to a /8. These sensors represent a range of organizations and a diverse sample of the routable IPv4 space including nine of all routable /8 address ranges. Data gathered from these deployments is used to demonstrate the ability of the IMS to capture and characterize several important Internet threats: the Blaster worm* (August 2003)*, the Bagle backdoor scanning efforts* (March 2004)*, and the SCO Denial of Service attacks* (December 2003).

1 Introduction

As national infrastructure becomes intertwined with emerging global data networks, the stability and integrity of the two have become synonymous. This connection, while necessary, leaves network assets vulnerable to the rapidly moving threats of today's Internet, including fast moving worms, distributed denial of service attacks, and routing exploits. These threats share several key properties. First and foremost these threats 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.

One promising method for investigating these threats is the monitoring of unused or dark address space [22, 11]. Because there are no legitimate hosts in an unused address block, traffic must be the result of misconfiguration, backscatter from spoofed source addresses, or scanning from worms and other probing. This pre-filtering provides an excellent method of studying Internet threats, however we believe there are two key design challenges that must be addressed when constructing a monitoring infrastructure based on this technique.

The first issue is sensor coverage. Sensor coverage refers to the visibility of the system into Internet threats. One method to increase visibility is to monitor larger blocks of address space [13]. The problem is that the IPv4 space is limited and there are a small number of large unused address blocks available for instrumentation. In addition, it has been shown that address blocks in different networks see different threat traffic [7]. Thus, sensor size and topological location are important components of sensor coverage.

The second issue is service emulation. If the sensors do not directly involve live hosts then there is a question of what services to emulate and at what level to emulate them. This is a difficult problem given the immense number services on the Internet today. An ideal system would reproduce all current and future services with exactly the same behavior as all possible end-hosts. Such a system is impossible, so there must a tradeoff.

The IMS is a distributed, globally scoped, Internet threat monitoring system designed with these challenges in mind. The goal of the IMS is to measure, characterize, and track a broad range of Internet threats. This means achieving the maximum possible sensor coverage with enough fidelity to gather intelligence on specific services. The fundamentally distributed nature of this architecture allows the IMS to monitor diverse addresses and topologies. In order get the most information possible given the scale of system, 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 service emulation. This approach is used in conjunction with an innovative payload signature and caching technique. Observations have shown that many of the packets seen at the sensors are duplicates, making this caching technique highly effective in managing the overhead of storing payloads.

The main contributions of this paper are:

- **The design and implementation of a distributed, globally scoped, Internet threat monitoring system.** The IMS architecture is based on three novel components. First, a *Distributed Monitoring Infrastructure* increases visibility into global threats. Second, a *Lightweight Active Responder* provides enough interactivity that traffic on the same service can be differentiated independent of application semantics. Third, a *Payload Signatures and Caching* mechanism avoids recording duplicated payloads, reducing overhead and assisting in the identification of new payloads.

- **The deployment and demonstration of the IMS on production networks.** The current IMS deployment consists of 28 monitored blocks at 18 physical installations. These deployments range in size from a /25 to a /8 and include major service providers, large enterprises, academic networks, and broadband providers. These sensors represent a range of organizations and a diverse sample of routable IPv4 space. Data gathered from these deployments is used to demonstrate the ability of the IMS to capture and characterize several important Internet threats.

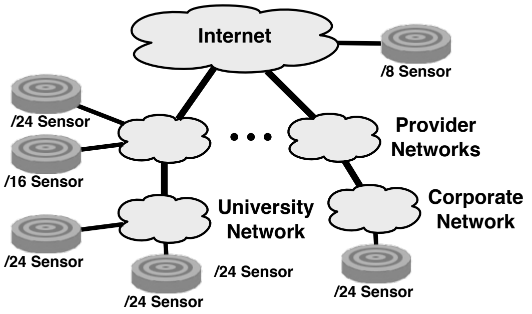


Figure 1. Internet Motion Sensor Architecture

The paper is organized as follows: Section 2 details the architecture and core components of the IMS. Section 3 describes the IMS deployment and highlights the inovative capaibilities of this system by examining three distinct events. Section 4 provide an overview of related work. Finally, Section 5 discusses future directions and conclusions.

2 Internet Motion Sensor Architecture

The Internet Motion Sensor was designed to measure, characterize, and track a broad range of Internet threats. This particular challenge necessitates a lightweight monitoring approach having global visibility. More specifically, the IMS was designed to:

- Maintain a level of interactivity that can differentiate traffic on the same service.

- Provide visibility into Internet threats beyond address, geographical, and operational boundaries.

- Enable characterization of emerging threats while minimizing incremental effort.

While other systems have combined sensors of different measurement fidelities [32], the goal of our project is to gain global threat visibility rather than in-depth information on the specific mechanisms of a threat. This implies that a simple system that captures less data may actually be more effective then a complex system producing large amounts of data. In other words, it is possible to trade in-depth threat information for the ability to gain additional visibility.

The tradeoff of visibility over in-depth threat information motivates the architecture described in this section. The IMS consist of a set of distributed blackhole sensors, each monitoring a dedicated range of unused IP address space. Because there are no legitimate hosts in an unused address block, traffic must be the result of misconfiguration, backscatter from spoofed source addresses, or scanning from worms and other probing. The blackhole sensors in the IMS have an active and a passive component. The passive component records packets sent to the sensor's address space and the active component responds to specific packets to elicit more data from the source.

The active component is designed to elicit the first payload of data across the major protocols (TCP, UDP, and ICMP). UDP is a connectionless protocols so application-level data is sent without the receiver ever responding. For example, the Witty [21] and Slammer [12] 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-oriented protocol and requires an active response to elicit any payload data. The IMS uses a simple lightweight active responder to establish a TCP connection and capture payload data on TCP worms like Blaster [16] and Sasser [8]. ICMP data is passively collected.

Storing the full payload for every packet has significant space requirements, 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 (signature), has already been recorded, the 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.

This architecture offers three novel contribibutions:

- **Distributed Monitoring Infrastructure:** The IMS is designed from the ground up for distributed deployment to increase visibility into global threats, including those that may illustrate targeting preferences.

- **Lightweight Active Responder:** The active responder is designed to maintain a level of interactivity that can differentiate traffic on the same service independent of application semantics. This enables IMS to characterize threats on emergent ports and services without additional deployments or scripting.

- **Payload Signatures and Caching:** The IMS checksums and caches commonly seen packets such that only new payloads need be stored. This saves significant storage resources and enables a simple mechanism for identifying new payloads.

The following three subsections describe and validate these novel components of the architecture by showing them in the context of a 28 address block, distributed IMS deployment.

2.1 Distributed Blackhole Network

A key contribution of the IMS is the creation of a large distributed sensor network built from address blocks of various sizes and placed in a variety of topologically diverse locations. While previous attempts at monitoring unused address block have focused on three or fewer address blocks [22, 11, 32, 17], our approach is to be widely distributed. This architectural choice has two key advantages; greater visibility into distant events and broader coverage of threats.

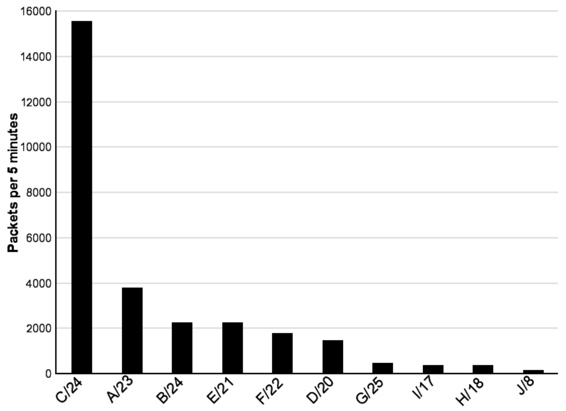


Figure 2. Average packet rate (5 minute bins) as seen by ten IMS Blackhole sensors over one month, normalized by /24

Using the analogy of Astrometric Telescopes, Moore in [11] notes that the greater the size of a light telescope, the greater the visibility into fainter, smaller, further and older objects. Moore notes that the more address space monitored (for random scanning events) the better the insight into shorter lived or lower rate events. The conclusion is that having large address blocks is important for monitoring globally scoped events. In addition to wide address blocks, Moore also suggests distributed blocks as a method for increasing visibility.

Distributed sensors provide more addresses that increase visibility and also another important benefit, broader coverage of threats. Previous work [7] has noted the surprising finding that dark address monitors can see strikingly different behaviors, even when local-preference scanning is removed.

As an example, consider the packet rate observed by a collection of sensors. Figure 2 shows the amount of traffic over all protocols and services observed by ten blackhole sensors. Packets are normalized by the size of a /24 so sensors covering different sized blocks can be compared. Normalization is performed such that the magnitude seen in a /23 would be divided by two and traffic in a /25 multiplied by two.

Figure 2 shows that the amount of traffic varies dramatically and can differ by more than two orders of magni-

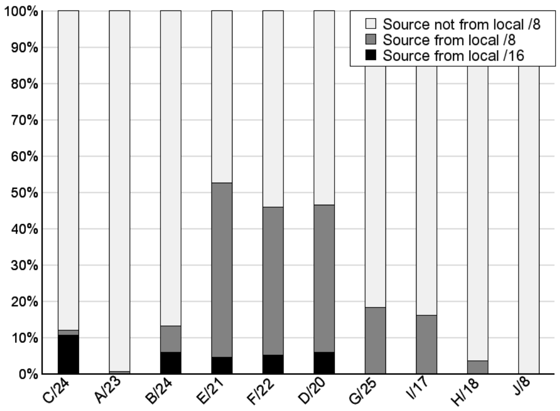


Figure 3. Distribution of local preference accross ten IMS blackhole sensors over one month

tude between sensors. The local target selection of malicious attackers or worms with local scanning preference in their scanning algorithms[26], such as Code Red II [3], Nimda [2], and Blaster [16] are likely canidates for the difference in magnitude. However, we show that this is not the case.

Figure 3 shows the percentage of traffic to all protocols and services from the same local /16 and local /8 as the sensor where the traffic was observed. There are two important implications of this graph. First, there are very different relative amounts of local /8 traffic seen at the various sensors. Second, although some sensors see a very significant amount of normalized local /8 traffic, those blocks do not always correlate with the sensors with the greatest magnitude of overall traffic. For example, C/24 observes by far the greatest amount of traffic, but only approximately 10% of that traffic is from within the same /8 as the sensor. So, even though local traffic can be significant, the major traffic differences are not due to local preference. Additional insights into these differences are discussed in [7]. Thus, the important message in Figure 2 and Figure 3 is that different blackholes observe different magnitudes and types of traffic.

The distributed nature of IMS affords us additional visibility into global threats. This comes from our ability to increase address space beyond a single wide address block as well as the topological and organizational diversity of the address blocks.

2.2 Lightweight Responder

Another novel aspect of this work is the construction of a lightweight responder whose main responsibility it is to elicit payloads for TCP connections. Recall that because TCP is a connection oriented protocol [27], no application data is sent until after connection establishment. This has two major repercussions for any TCP based threats; threats to the same port can not be distinguished from each other, and threats will not send the exploit payload if a connection can not be established.

As an illustration, consider the Blaster worm [16]. The infection and transmission method used by Blaster is relatively complicated compared to a single-packet worm like Slammer [12]. A condensed diagram of the transactions involved in a Blaster infection is illustrated in Figure 4(a). 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.

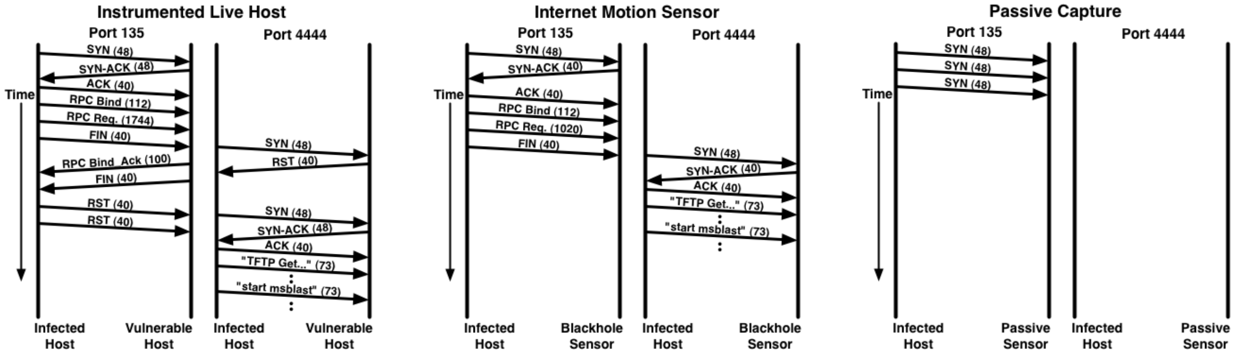
Figure 4(b) depicts the transactions of the Blaster worm as captured by a blackhole sensor in the IMS. Observe how a single SYN-ACK on port 135 elicits not only the exploit, but also a SYN on the backdoor port. Since the IMS blackhole sensors respond to SYN packets on all ports, the worm connects to port 4444 and sends commands to TFTP the payload and start the worm binary. Compare the data captured by the IMS to the data recorded by a passive blackhole monitor, shown in Figure 4(c). 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 illustrates the two key contributions of the lightweight reposnder; the ability to elicit payloads to differentiate traffic and the ability to get responses across ports without application semantic information. The following subsections explores these points in more depth.

2.2.1 Differentiate Services

An important illustration of the value in having more information is the extraction and classification of a new threat for a highly trafficked service. The Sasser [8] worm utilized TCP port 445, which is a used by 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, shown in Figure 5. Figure 5(a) shows the traffic captured on port 445 over the period of 7 days. Figure 5(b) 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 by using payload signatures.

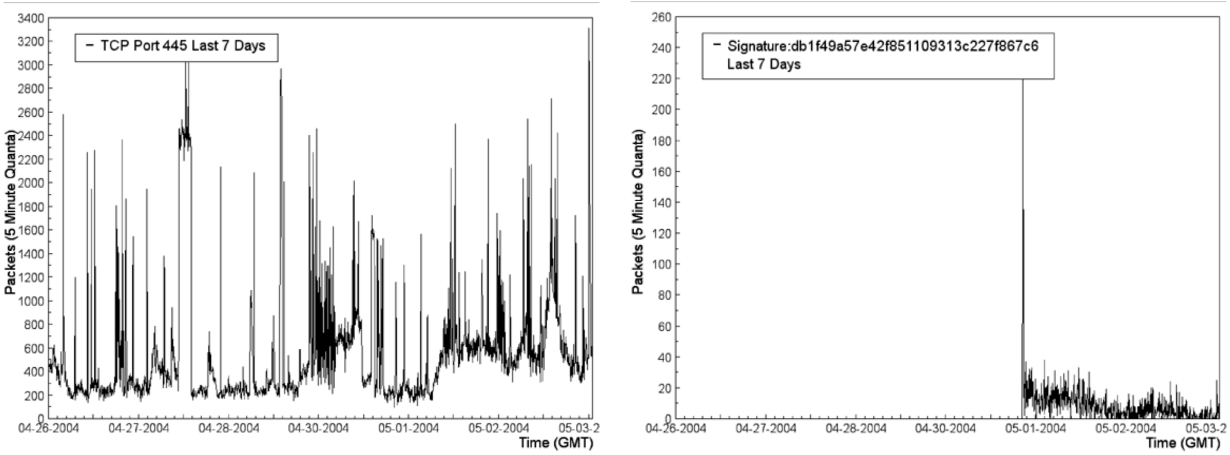
2.2.2 Service Agnostic

Another advantage of the lightweight responder is its service agnostic approach which enables insight into less popular services. Consider, for example, a management appli-



(a) (b) (c)

Figure 4. Blaster infection attempt captured using three monitoring techniques



(a) (b)

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

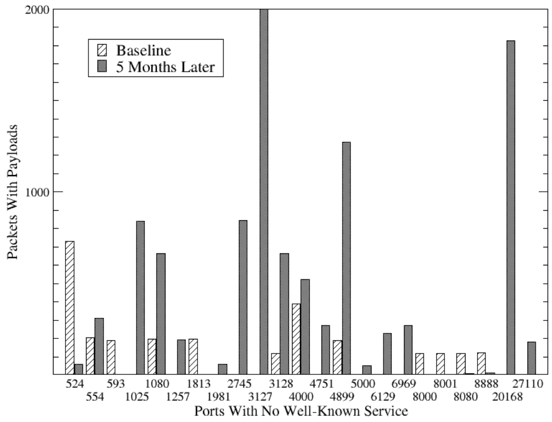


Figure 6. Change in activity on TCP ports without well-known services

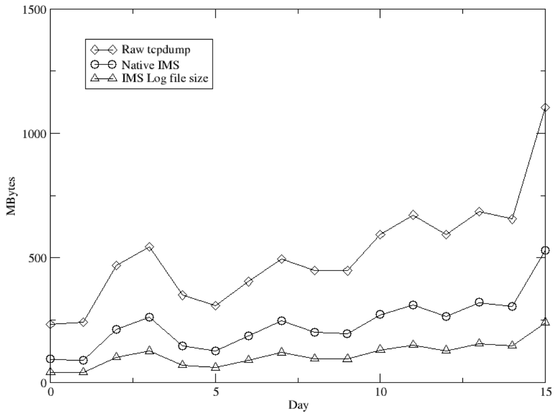


Figure 7. IMS and tcpdump log file sizes over a 16 day period

cation 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. Consider Figure 6, which shows traffic on TCP ports which do not have well known services. This figure shows the top 20 ports which had significant changes in traffic levels over a five month period. Note in particular, ports 2745 and 3127, which represent services or backdoors as discussed above.

2.2.3 Limitations

The service agnostic lightweight responder is an novel method of tracking emerging threats, however, it may provide little or no information on the threats that depend on application level responses. For example, the NetBIOS service which runs on Windows systems requires an RPC *bind()* before performing a RPC *request()* to execute many popular operations. Thus, an IMS sensor may observe the RPC *bind()* but not the subsequent RPC *request()* because no application level response was ever sent. However, in some cases the threat will continue without waiting for the application level response. For example, the Blaster worm will send the RPC *bind()* and RPC *request()* (containing the exploit) without any application level response. In addition, many threats perform operations on multiple ports and it is possible to track these threats by observing the pattern of requests across ports.

While responses across most or all ports provides insights into potentially unknown threats, we would be remiss if we did not point out that this makes these sensors simple to identify and fingerprint. One main advantage of this system is its focus on globally scoped threats over the activites for individual attackers. This translates into a threat model whereby our biggest concern is the encoding of the monitored network blocks in threat "no-hit-lists." The large number of distributed blocks of small to medium size makes this proposition difficult. We have not seen any evidence to date of this type of activity being encoded into self propgrating threats. Nevertheless, we are exploring several ways of discouraging fingerprinting including; sensor rotation (i.e. continuously moving the active responders to evade detection), source squelching on individual sensors [17] or accross the entire system (i.e. if we detect and respond at one sensor, we don't have to send a response from other sensors), or building simulated hosts and topology (as in honeyd [19]) to mask the presence of a blackhole.

2.3 MD5 Checksuming and Caching of request payloads

The final novel aspect of this system is its method of storing payloads. 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. 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. This approach offers a factor of two savings in disk (Figure 7) and the hit rate on the signature cache typically tops 96% (Figure 8). The implication is that a large number of

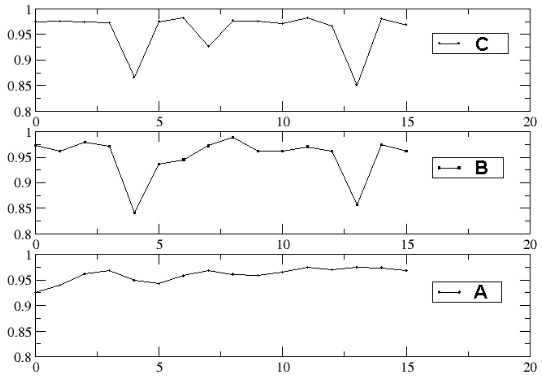


Figure 8. Signature database hit rate over a 16 day period on three blackhole sensors. Fluctuations in the hit rate in two of the sensors correspond with an attack on a MySQL Server service

the payloads seen each day are exactly the same. It is surprising that a cache hit rate of greater than 96% only yields a factor of two increase in savings, however, the payload distribution is heavily skewed to packets with little or no payload.

The factor of two in storage savings helps significantly since traces gathered on a /8 can reach over 100GB/day. In addition, checksuming every payload provides an efficient signature system. For example, the signatures can be used to generate simple metrics for the amount of new data observed or as a first pass to a more expensive string matching algorithm. In summary, the MD5 checksuming and caching system in IMS provides a factor of two in storage savings and a simple, fast signature mechanism.

3 Deployment Observations and Experiences

This section describes the realization of the IMS architecture. We describe the current distributed IMS deployment consisting of 28 distinct address blocks at 18 physical installations. In addition, we demonstrate the ability of the IMS to capture and characterize several important Internet threats that were previously not possible or difficult using passive monitoring methods. Three threat events captured using the IMS deployment are presented to show how the lightweight responders, payload signatures and caching, and the distributed nature of the IMS enable novel types of globally scoped analyses.

1. Internet Worms: The IMS can differentiate scan traffic from real worms on TCP ports allowing it to capture and study TCP worms like Blaster.

2. Scanning: Because the IMS doesn't rely on service modules, it can detect scanning activity targeted at less well know services like worm/virus backdoors.

3. DDoS: The distributed architecture enables observations of DDOS events including those that may not generate much traffic or target a wide range of addresses.

3.1 Distributed Deployment

One key contribution of the IMS is its ability to support a widely distributed deployment. The current deployment consist of 28 distinct monitored blocks at 18 physical installations (see Table 3.1). These deployments range in size from a /25 to a /8 and include major service providers, large enterprises, academic networks, and broadband providers. These sensors represent a range of organizations and a diverse sample of the routable IPv4 space including nine of all routable /8 address ranges.

The first version of the IMS was deployed in 2001 as part of a global threat monitoring system developed by researchers at Arbor Networks and the University of Michigan. This initial installation was built to monitor a /8 network, or approximately 1/256th of the total IPv4 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 the system was updated and expanded to represent the distributed architecture presented in this paper.

The current IMS revision contains lightweight responders across all ports and a data query engine to support distribution. The query system consists of a query daemon that is installed on each sensor and listens for queries from a portal server. The portal server acts as a query aggregator forwarding queries to each sensor and forwarding back the responses. The portal server also runs a web application that utilizes the query system to provide real-time views of threat traffic in the IMS sensor network. While, the query system is an interesting implementation on top of the IMS architecture, it is beyond the scope of this paper.

3.2 Internet Worms

Internet worms represent a class of security threats that seek to execute code on a target machine by exploiting vulnerabilities in the operating system or application software [15, 33]. Unlike viruses, however, worms do not rely on attaching themselves to files to propagate. Rather, worms stand-alone and propagate by using the network to scan for other potentially vulnerable hosts and then exploit them without user interaction.

Organization	Size
Academic Network	/24, /24
Academic Network	/24
Academic Network	/24
Academic Network	/24
Large Enterprise	/18
Tier 1 ISP	/17
Tier 1 ISP	/18
National ISP	/20, /21, /22
National ISP	/24, /24, /24
National ISP	/24
National ISP	/24
National ISP	/24, /24
Regional ISP	/25, /24, /24
Regional ISP	/23, /22
Regional ISP	/8
Regional ISP	/24
Broadband Provider	/17, /18, /22
Broadband Provider	/24, /24

Table 1. IMS Deployments

Globally scoped network monitoring systems, such as the IMS, are helpful in characterizing, measuring and tracking 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 the type of analysis made possible by the IMS, 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 generate an address to infect (60% were randomly generated and 40% were located within the same host /16 network as the affected system). The worm would then sequentially scan from the chosen address. The IMS was able to measure the release and propagation of the Blaster worm as it attempted to infect random hosts.

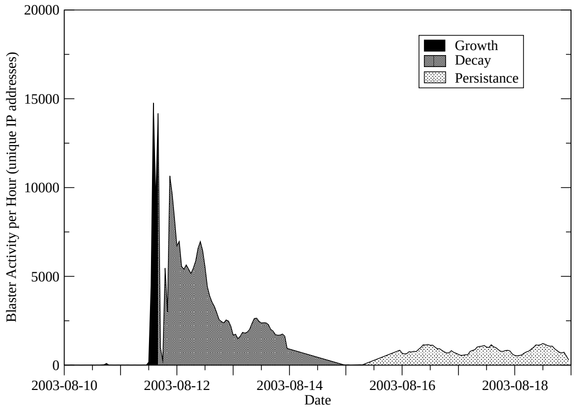


Figure 9. A snapshot of Blaster worm showing the three phases of the worm lifecycle

The 7-day period of observations surrounding the release of the Blaster worm indicates a clear 3-phased cycle. Figure 9 illustrates the three phases using the number SYNs to TCP port 135 on a /8 (the active responder was not yet operational). The first worm phase is the growth phase in which the number of scans increased from a baseline rate to hundreds of thousands per hour. The second phase is the decay phase in which in the number of observed probes drops as large-scale filtering was implemented to halt the worm's spread and cleanup started. The third phase of worm activity is the persistence phase which for the Blaster worm has continued through 2004.

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.

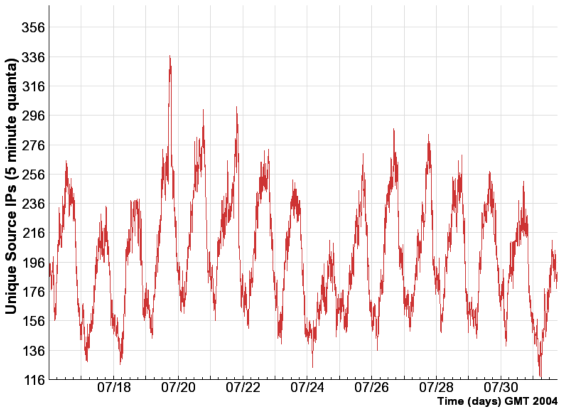


Figure 10. Circadian pattern in unique Blaster hosts over a 7 day period in July 2004

Continued Blaster activity displays a circadian pattern, with peak activity occurring near 05:00 GMT as shown in Figure 10. The unique Blaster IPs shown in the graph are identified as Blaster hosts but a command sequence specific to Blaster on the Blaster backdoor port on TCP port 4444. Thus a host is only identified as a Blaster host if it first sends the exploit on TCP port 135 and then follows with the command sequence on TCP port 4444. Figure 10 shows between 100 and 350 unique Blaster hosts per 5 minute period over a 7 day period in July 2004. This pattern is typical of Blaster traffic observed since its inception. The 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.

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.

The above Blaster analysis illustrates how IMS provides additional insight into highly salient threats. Blaster utilizes TCP as a transport protocol and thus attempting to track and characterize it using a passive blackhole monitor is difficult or more often impossible. A passive monitor will detect the SYN packet but not the worm payload making it very hard to differentiate the worm traffic from scans and other traffic on that port. In addition, because IMS gets the first payload, and often traffic on followup ports (like port TCP port 4444 for Blaster), IMS provides enough data to differentiate between the different variant of worms. Differentiating between the variants of Blaster was possible using the command packets received on TCP port 4444. This type of analysis is not possible with a passive blackhole monitor.

3.3 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 autorooters, 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. The IMS has the ability to respond as if these new services were running, allowing the collection of the scan payload. An interesting application of this data 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 on backdoor ports left by widespread variants of the Bagle [30] and MyDoom [1] mail-based worms. The patterns of sources for this traffic show that they are widespread. The payloads identified suggest that these hosts may be under attack from another worm such as AgoBot [31] or opportunistic attackers such as those looking to build armies of compromised computers.

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 control 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 to execute arbitrary commands.

Figure 11 shows the scan traffic for 2745/TCP and 3127/TCP over a one week period starting March 20th of 2004 as observed by one /24 IMS Sensor. Scans against other ports open by the Bagle variants (including 6777 for

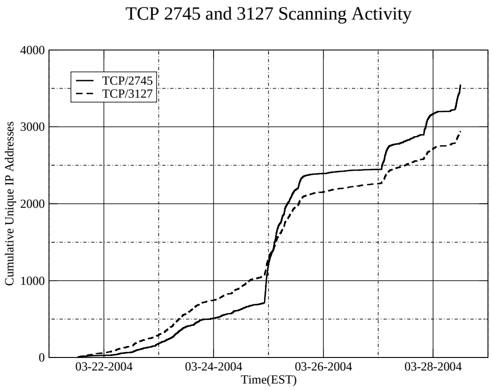


Figure 11. Propagation of 2745/TCP and 3127/TCP scanning. This figure shows the cumulative number of source IP addresses contacting these ports over a one week period as observed by one /24 sensor

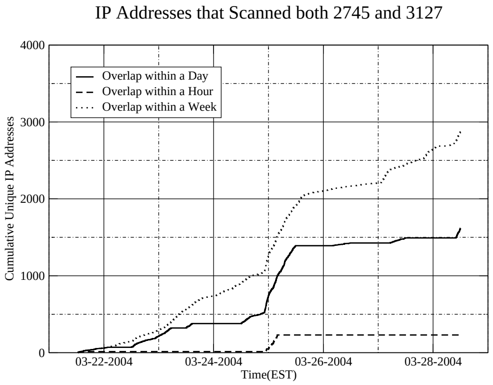


Figure 12. IP address overlap between 2745/TCP and 3127/TCP sources. The three datasets show the addresses that are considered overlapping if they scan both ports in the same hour, the same day, or the same week

Bagle.A, 8866 for Bagle.B, and 11117 for Bagle.L) have not been observed in any appreciable quantities. In addition, the IMS has not observed significant activity in this time period against additional MyDoom ports, including TCP ports 3128-3198.

While the similiar magnitude and shape of the datasets in Figure 11 may suggest that these scans are occuring concurently from the same souorce addresses, this is actually not the case. Figure 12 shows the cumulative unique source addresses that scanned both 2745/TCP and 3127/TCP. The three datasets represent three distinct definitions of overlapping; source addresses that scan both ports within the same hour, within the same day, and within the same week. This view shows us two interesting facts. First sources that scanned one port did not scan the other port in the same hour indicating that these were not lock step scans. Secondly, when observered over the course of a week, nearly all sources scanned both ports, indicating that while the target selections were independent of each other, the same pool of source addresses were being used for both scans.

```
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

The top payload captured using IMS on port 2745/TCP (backdoor ports left by the Bagle.C-G and Bagle.J-K mail-based worms) is shown in Figure 13. Note that the signature in Figure 13 differs from the Bagle removal mechanism described by Joe Stewart [28]. The similarity of the first four bytes (0x43 0xff 0xff 0xff) must be noted; after that point the signatures are divergent. This may indicate a common command or authentication byte sequence.

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 activity on the Bagle and MyDoom backdoor ports captured by the IMS clearly demonstrates the value of the lightweight responder. Without the ability to respond as if these new services were running, the IMS would never have been able to collect the payload and analyze this activity. This event also shows the advantage of using a simple lightweight responder over handcrafted service modules. Because the backdoors observed were from relatively new worms having a huge number of variants, it would be extremely time consuming to create services modules for each variant in time to capture these events.

3.4 Distributed Denial of Service Attacks

The final event presented here exhibits how IMS has visibility into 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

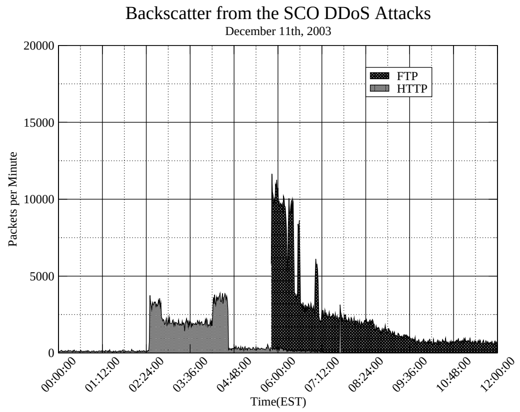


Figure 14. The two largest events of the December 2003 SCO DDoS attacks.

Table 2. Discrete DDoS events targeting SCO

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

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). The two largest components of this attack can be seen in Figure 14. 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. 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.

The events depicted in Table 2 illustrate that DDoS events can be long lived and made up of several smaller events that can be combined to create larger attacks. Furthermore, this event 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 to 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%).

4 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.

Antivirus software [5] and Host Based intrusion detection systems [9] 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 [4, 6] track threats by providing an exploitable resource and monitoring it for abnormal behavior. A major goal of honeypots [24] is to provide insight into motivation and techniques behind these threats. While these host based monitoring techniques can provide detailed information on threats, they are limited in their ability to scale to monitor large address blocks.

The alternative to host based monitoring techniques is to monitor from the network perspective. By far the most common technique is the passive measurement of live networks. In contrast to active techniques, passive network monitoring approaches attempt not to intrude on the existing operation of the network. This can be accomplished by monitoring data from existing security or policy enforcement devices. By either watching firewall logs, looking for policy violations, or by aggregating IDS alerts across multiple enterprises [20, 34], 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 [29] 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 [10] offer another readily available source of information. While monitoring live networks provides broader coverage, these techniques often face difficulties in distinguishing between benign and malicious traffic.

Another interesting network monitoring approach is to passively collect data from traffic to unused (or dark) address space. Because this space has no legitimate hosts, traffic destined to the space 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 [11, 14, 23]. In contrast to host based techniques, passively monitoring unused address space is able to scale to very large address spaces at the expense of not gathering details about specific events.

The final 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 [25] and iSink [32], and software like honeyd [19] are used to bring up networks of honeypots; places designed to capture information about intrusions in a controlled environments. These techniques attempt to balance the scalability of dark address monitors while gathering more detailed information about threats.

The techniques described above provide varying amounts of intelligence regarding a threat. Some systems capture 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 [11, 32, 23]. 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 [12, 21]. However, given the finite size of the IPv4 address space, it is important to explore new methods of obtaining breadth. The IMS addresses the issue of breath by creating a distributed architecture consisting of numerous topologically diverse sensors.

Depth defines the extent to which a sensor emulates the services and characteristic of a real host, similar to the interaction spectrum in honeypots [24]. 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.* [32]. The IMS, however, uses an extension to passive techniques [11] that gains additional intelligence without emulating services to the extent of previous active methods [25, 19].

5 Conclusion

This paper describe the Internet Motion Sensor, a distributed, globally scoped, Internet threat monitoring system. With roots in wide, dark address monitoring, the IMS extends these techniques to include a distributed blackhole network with a lightweight responder and a novel payload signature and caching mechanism. Together, these capabilities afford new insight into Internet worms, denial of service attacks, and malicious scan activity.

First, we presented the IMS architecture and highlighted the innovative capacities of the system. The distributed blackhole network allows increased visability into more distant events. The lightweight responder is designed to differentiate traffic services while remaining independent of application semantics. Finally, the payload signature and caching mechanism reduces the overhead associated with storing request payloads.

We then presented this architecture in the context of a 28 block, 18 organization distributed deployment. Highlighting the innovative capabilities of IMS, we examined three distinct large-scale events including, the Blaster worm (August 2003), the Bagle backdoor scanning (March 2004), and the SCO denial of service attacks (December 2003).

There remain several interesting issues that would enhance the capabilities of the IMS. First, we are interested in the issue of fingerprinting the blackhole sensor and inclusion in blacklists. We wish to examine several of the anti-fingerprinting techniques discussed in Section 2. Second, we wish to further explore the tradeoffs between breadth and depth of threat monitoring architectures. The most promising improvement being the creation of a hybrid system that combines host-based sensors with wide address space monitors. Finally, we hope to develop additional techniques for characterizing attackers. Techniques such as passive OS fingerprinting [18] and firepower calculations will provide additional information about the scope and impact of an ongoing attack.

Acknowledgments

This work was supported by the Advanced Research and Development Activity (ARDA) under contract number NBCHC030104.

The authors would like to thank all the IMS participants for their help and suggestions. We would also like to thank Arbor Networks and Larry Blunk, Bert Rossi, and Manish Karir at Merit Network for their assistance and support through this paper. The IMS project had its roots in an earlier system created by Dug Song, Robert Stone, and G. Robert Malan.

References

[1] CERT. Incident Note IN-2004-01: W32/Novarg.A Virus. `http://www.cert.org/incident_notes/IN-2004-01.html`, 2004.

[2] CERT Coordination Center. CERT Advisory CA-2001-26 Nimda Worm. 2001.

[3] CERT Coordination Center. Code Red II: Another Worm Exploiting Buffer Overflow In IIS Indexing Ser-

vice DLL. Available at http://www.cert.org/incident_notes/IN-2001-09.html, 2001.

[4] B. Cheswick. An evening with Berferd in which a cracker is lured, endured, and studied. In *Proceedings of the Winter 1992 USENIX Conference: January 20 — January 24, 1992, San Francisco, California*, pages 163–174, Berkeley, CA, USA, Winter 1992.

[5] F. Cohen. *A Short Course on Computer Viruses*. John Wiley & Sons, 2nd edition, April 1994.

[6] F. Cohen. The deception toolkit (DTK). http://www.all.net/dtk/dtk.html, June 2004.

[7] E. Cooke, M. Bailey, Z. M. Mao, D. Watson, and F. Jahanian. Toward understanding distributed blackhole placement. In *Proc of ACM CCS Workshop on Rapid Malcode*, pages 54–64. ACM Press, October 2004.

[8] M. Corporation. What you should know about the Sasser worm. http://www.microsoft.com/security/incident/sasser.mspx, May 2004.

[9] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff. A sense of self for Unix processes. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 120–128, Oakland, CA, May 1996.

[10] C. S. Inc. Netflow services and applications. 2002. http://www.cisco.com/warp/public/cc/pd/iosw/ioft/neflct/tech/napps_wp.htm.

[11] D. Moore. Network telescopes: Observing small or distant security events. In *11th USENIX Security Symposium, Invited talk*, San Francisco, CA, Aug. 5–9 2002. Unpublished.

[12] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. Inside the Slammer worm. *IEEE Security & Privacy*, 1(4):33–39, 2003.

[13] D. Moore, C. Shannon, G. M. Voelker, and S. Savage. Network telescopes. http://www.caida.org/outreach/papers/2004/tr-2004-04/, July 2004.

[14] D. Moore, G. M. Voelker, and S. Savage. Inferring Internet denial-of-service activity. In *Proceedings of the Tenth USENIX Security Symposium*, pages 9–22, Washington, D.C., Aug. 13–17 2001.

[15] J. Nazario. *Defense and detection strategies against Internet worms*. Artech, 2004.

[16] J. Nazario, M. Bailey, and F. Jahanian. The spread of the Blaster worm. Submitted for publication.

[17] R. Pang, V. Yegneswaran, P. Barford, V. Paxson, and L. Peterson. Characteristics of internet background radiation. http://www.cs.princeton.edu/nsg/papers/telescope.pdf, June 2004.

[18] T. H. Project. Know Your Enemy: Passive Fingerprinting, Identifying remote hosts, without them knowing. 2002.

[19] N. Provos. Honeyd — A virtual honeypot daemon. In *10th DFN-CERT Workshop*, Hamburg, Germany, Feb. 2003.

[20] SANS. Sans - Internet storm center - cooperative cyber threat monitor and alert system, June 2004.

[21] C. Shannon and D. Moore. The spread of the Witty worm. http://www.caida.org/analysis/secuirty/witty/, June 2004.

[22] D. Song, R. Malan, and R. Stone. A snapshot of global Internet worm activity. FIRST Conference on Computer Security Incident Handling and Response 2002, June 2002.

[23] D. Song, R. Malan, and R. Stone. A snapshot of global Internet worm activity. Arbor Network Technical Report, June 2002.

[24] L. Spitzner. *Honeypots: Tracking Hackers*. Addison-Wesley, 2002.

[25] L. Spitzner et al. The honeynet project. http://project.honeynet.org/, June 2004.

[26] S. Staniford, V. Paxson, and N. Weaver. How to 0wn the Internet in your spare time. In *Proceedings of the 11th USENIX Security Symposium*. USENIX, Aug. 2002.

[27] W. R. Stevens. *TCP/IP Illustrated, Volume 1: The Protocols*. Addison Wesley, 1994.

[28] J. Stewart. [full-disclosure] [fwd: [th-research] bagle remote uninstall]. http://lists.netsys.com/pipermail/full-disclosure/2004-January/015970.html, January 2004.

[29] R. Stone. CenterTrack: An IP overlay network for tracking DoS floods, 2000.

[30] Symantec. Security Response - W32.Beagle.A. http://securityresponse.symantec.com/avcenter/venc/data/w32.beagle.a@mm.html, 2004.

[31] I. Trend Micro. WORM AGOBOT.GEN - Description and solution. http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=WORM_AGOBOT.GEN, 2003.

[32] P. B. Vinod Yegneswaran and D. Plonka. On the design and use of Internet sinks for network abuse monitoring. In *Recent Advances in Intrusion Detection—Proceedings of the 7th International Symposium (RAID 2004)*, Sophia Antipolis, French Riviera, France, Oct. 2004.

[33] N. Weaver, V. Paxson, S. Staniford, and R. Cunningham. A Taxonomy of Computer Worms. In *Proc of ACM CCS Workshop on Rapid Malcode*, pages 11–18. ACM Press, October 2003.

[34] V. Yegneswaran, P. Barford, and S. 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.