[1] M. Abdelhafez and G. Riley. Evaluation of worm containment algorithms and their effect on legitimate traffic. In *Third IEEE International Workshop on Information Assurance (IWIA)*, March 2005. [bib]

[2] K. G. Anagnostakis, M. B. Greenwald, S. Ioannidis, A. D. Keromytis, and D. Li. A cooperative immunization system for an untrusting Internet. In *Proceedings of the 11th IEEE International Conference on Networks (ICON'03)*, October 2003. [bib]

[3] M. Atighetchi, P. Pal, F. Webber, R. Schantz, C. Jones, and J. Loyall. Adaptive cyberdefense for survival and intrusion tolerance. *IEEE Internet Computing*, 8(6):25-33, November/December 2004. [bib | DOI]

> While providing some resistance against cyberattacks, current approaches to securing networked and distributed information systems are mainly concerned with static prevention measures. For example, signature-based systems can only detect known attacks and tend to provide brittle, all-or-nothing protection. New work in survivability and intrusion tolerance focuses on augmenting existing information systems with adaptive defenses. A middleware-based survivability toolkit lets applications use network- and host-based mechanisms in their own defense.
>
> Keywords: Homeland security, faul-tolerance, intrusion detection, middleware

[4] R. Bajcsy, T. Benzel, M. Bishop, B. Braden, C. Brodley, S. Fahmy, S. Floyd, W. Hardaker, A. Joseph, G. Kesidis, K. Levitt, B. Lindell, P. Liu, D. Miller, R. Mundy, C. Neuman, R. Ostrenga, V. Paxson, P. Porras, C. Rosenberg, J. D. Tygar, S. Sastry, D. Sterne, and S. F. Wu. Cyber defense technology networking and evaluation. *Communications of the ACM*, 47(3):58-61, 2004. [bib | DOI]

[5] R. Browne. C4I defensive infrastructure for survivability against multi-mode attacks. In *Proceedings of the 21st Century Military Communications Conference (MILCOM)*, volume 1, pages 417-424, October 2000. [bib | DOI]

> A previous paper points out that the United States and her allies cannot achieve information superiority simply by prevailing at information warfare. 21st century C4I systems must be able to defend against “multi-mode” attacks, which are enemy strategies using clever combinations of conventional and non-conventional warfare. Owing to the problem of multi-mode attacks, completely new approaches to C4I defensive architecture are needed. This current paper criticizes some popular 20th century C4I defense technologies, such as adaptive autonomic defenses and encapsulated self-healing networks and systems, all of which are technologies with severe inherent weakness against multi-mode attacks. This paper is a speculative discussion of new C4I defense technologies as well as policy issues regarding information superiority that have never been adequately addressed in the literature. The intent is to stimulate new research and development to the benefit of practical fielded C4I systems.

[6] D. Brumley, L.-H. Liu, P. Poosankam, and D. Song. Design space and analysis of worm defense strategies. In *Proceedings of the 2006 ACM Symposium on Information, Computer, and Communication Security (ASIACCS 2006)*, March 2006. [bib | .pdf]

> We give the first systematic investigation of the design space of worm defense system strategies. We accomplish this by providing a taxonomy of defense strategies by abstracting away implementation-dependent and approach-specific details and concentrating on the fundamental properties of each defense category. Our taxonomy and analysis reveals the key parameters for each strategy that determine its effectiveness. We provide a theoretical foundation for understanding how these parameters interact, as well as simulation-based analysis of how these strategies compare as worm defense systems. Finally, we offer recommendations based upon our taxonomy and analysis on which worm defense strategies are most likely to succeed. In particular, we show that a hybrid approach combining Proactive Protection and Reactive Antibody Defense is the most promising approach and can be effective even against the fastest worms such as hitlist worms. Thus, we are the first to demonstrate that it is possible to defend against the fastest worms such as hitlist worms.

[7] M. Cai, K. Hwang, Y.-K. Kwok, S. Song, and Y. Chen. Collaborative Internet worm containment.

IEEE Security and Privacy Magazine, 3(3):25-33, May/June 2005. [bib | DOI]

Large-scale worm outbreaks that lead to distributed denial-of-service (DDoS) attacks pose a major threat to Internet infrastructure security. Fast worm containment is crucial for minimizing damage and preventing flooding attacks against network hosts.

[8] B. Carrier, S. Jeyaraman, and S. Sellke. Impact of network design on worm propagation. Technical report, CERIAS. CERIAS TR 2004-35. [bib | .pdf]

In this paper, we simulate the Code Red II and Nimda worms on different enterprise-scale networks to determine the impact that topology has on worm propagation. A corporate network can be designed to improve security and, as we show, to decrease the propagation rate of worms that use network scanning as a target discovery technique. We also examine the impact that LaBrea-like devices have on propagation rates and compare it to the impact of network topology.

[9] S. Cheetancheri, D. Ma, T. Heberlein, and K. Levitt. Towards an infrastructure for worm defense evaluation. In *Proceedings of the 25th International Performance Computing and Communications Conference (Workshop on Malware)*, pages 559-566, April 2006. [bib]

[10] S. Chen and S. Ranka. An internet-worm early warning system. In *Proceedings of the IEEE Globecom 2004 - Security and Network Management*, volume 4, pages 2261-2265, November 2004. [bib | .html | .pdf]

We propose an Internet-worm early warning system, which integrates a set of novel techniques that automatically detect the concerted scan activity of an on-going worm attack. It is able to issue warning at the early stage of worm propagation and to provide necessary information for security analysts to control the damage. The system monitors a “used” address space. Unlike the traditional approach that keeps track of SYN packets, it relies on RESET packets to find the scan sources, which has greater accuracy and less overhead. The system is resilient to anti-monitor measures. Particularly, a sophisticated protocol is designed to distinguish faked scan sources from real scan sources. We provide an analytical study on the properties and effectiveness of this early warning system, and back up our claims by numerical results.

[11] S. Chen and Y. Tang. Slowing down internet worms. In *Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS)*, pages 312-319. IEEE Computer Society, March 2004. [bib | .html]

An Internet worm automatically replicates itself to vulnerable systems and may infect hundreds of thousands of servers across the Internet. It is conceivable that the cyber-terrorists may use a wide-spread worm to cause major disruption to our Internet economy. While much recent research concentrates on propagation models, the defense against worms is largely an open problem. We propose a distributed anti-worm architecture (DAW) that automatically slows down or even halts the worm propagation. New defense techniques are developed based on behavioral difference between normal hosts and worm-infected hosts. Particulary, a worm-infected host has a much higher connection-failure rate when it scans the Internet with randomly selected addresses. This property allows DAW to set the worms apart from the normal hosts. We propose a temporal rate-limit algorithm and a spatial rate-limit algorithm, which makes the speed of worm propagation configurable by the parameters of the defense system. DAW is designed for an Internet service provider to provide the anti-worm service to its customers. The effectiveness of the new techniques is evaluated analytically and by simulations.

[12] E. Cooke and F. J. D. Mcpherson. The zombie roundup: Understanding, detecting, and disrupting botnets. In *Proceedings of the Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI)*, pages 39-44, June 2005. [bib | .html | .pdf]

Global Internet threats are undergoing a profound transformation from attacks designed solely to disable infrastructure to those that also target people and organizations. Behind these new attacks is a large pool of compromised hosts sitting in homes, schools, businesses, and governments around the world. These systems are infected with a **bot** that communicates with a bot **controller** and other

bots to form what is commonly referred to as a **zombie army** or **botnet**. Botnets are a very real and quickly evolving problem that is still not well understood or studied. In this paper we outline the origins and structure of bots and botnets and use data from the operator community, the Internet Motion Sensor project, and a honeypot experiment to illustrate the botnet problem today. We then study the effectiveness of detecting botnets by directly monitoring IRC communication or other command and control activity and show a more comprehensive approach is required. We conclude by describing a system to detect botnets that utilize advanced command and control systems by correlating secondary detection data from multiple sources.

Keywords: botnets, honeypots, security

[13] M. Costa, J. Crowcroft, M. Castro, A. Rowstron, L. Zhou, L. Zhang, and P. Barham. Vigilante: end-to-end containment of Internet worms. In *Proceedings of the 20th ACM Symposium on Operating Systems Principles (SOSP)*, pages 133-147, October 2005. [bib | DOI]

Worm containment must be automatic because worms can spread too fast for humans to respond. Recent work has proposed network-level techniques to automate worm containment; these techniques have limitations because there is no information about the vulnerabilities exploited by worms at the network level. We propose Vigilante, a new end-to-end approach to contain worms automatically that addresses these limitations. Vigilante relies on collaborative worm detection at end hosts, but does not require hosts to trust each other. Hosts run instrumented software to detect worms and broadcast self-certifying alerts (SCAs) upon worm detection. SCAs are proofs of vulnerability that can be inexpensively verified by any vulnerable host. When hosts receive an SCA, they generate filters that block infection by analysing the SCA-guided execution of the vulnerable software. We show that Vigilante can automatically contain fast-spreading worms that exploit unknown vulnerabilities without blocking innocuous traffic.

[14] J. R. Crandall and F. T. Chong. Minos: Control data attack prevention orthogonal to memory model. In *Proceedings of the 37th International Symposium on Microarchitecture (MICRO)*, pages 221-232, 2004. [bib | DOI]

We introduce Minos, a microarchitecture that implements Biba's low-water-mark integrity policy on individual words of data. Minos stops attacks that corrupt control data to hijack program control flow but is orthogonal to the memory model. Control data is any data which is loaded into the program counter on control flow transfer, or any data used to calculate such data. The key is that Minos tracks the integrity of all data, but protects control flow by checking this integrity when a program uses the data for control transfer. Existing policies, in contrast, need to differentiate between control and non-control data a priori, a task made impossible by coercions between pointers and other data types such as integers in the C language. Our implementation of Minos for Red Hat Linux 6.2 on a Pentium-based emulator is a stable, usable Linux system on the network on which we are currently running a web server. Our emulated Minos systems running Linux and Windows have stopped several actual attacks. We present a microarchitectural implementation of Minos that achieves negligible impact on cycle time with a small investment in die area, and minor changes to the Linux kernel to handle the tag bits and perform virtual memory swapping.

[15] J. R. Crandall, Z. Su, S. F. Wu, and F. T. Chong. On deriving unknown vulnerabilities from zero-day polymorphic and metamorphic worm exploits. In *Proceedings of the 12th ACM Conference on Computer and Communications Security (CCS)*, pages 235-248, November 2005. [bib | DOI]

Vulnerabilities that allow worms to hijack the control flow of each host that they spread to are typically discovered months before the worm outbreak, but are also typically discovered by third party researchers. A determined attacker could discover vulnerabilities as easily and create zero-day worms for vulnerabilities unknown to network defenses. It is important for an analysis tool to be able to generalize from a new exploit observed and derive protection for the vulnerability.Many researchers have observed that certain predicates of the exploit vector must be present for the exploit to work and that therefore these predicates place a limit on the amount of polymorphism and metamorphism available to the attacker. We formalize this idea and subject it to quantitative analysis with a symbolic execution tool called DACODA. Using DACODA we provide an empirical analysis of 14 exploits (seven of them actual worms or attacks from the Internet, caught by Minos with no prior knowledge of

the vulnerabilities and no false positives observed over a period of six months) for four operating systems.Evaluation of our results in the light of these two models leads us to conclude that 1) single contiguous byte string signatures are not effective for content filtering, and token-based byte string signatures composed of smaller substrings are only semantically rich enough to be effective for content filtering if the vulnerability lies in a part of a protocol that is not commonly used, and that 2) practical exploit analysis must account for multiple processes, multithreading, and kernel processing of network data necessitating a focus on primitives instead of vulnerabilities.

[16] J. R. Crandall, S. F. Wu, and F. T. Chong. Experiences using minos as a tool for capturing and analyzing novel worms for unknown vulnerabilities. In *Proceedings of the Second International Conference on Intrusion and Malware Detection and Vulnerability Assessment (DIMVA)*, pages 32-50, July 2005. [bib | DOI]

We present a honeypot technique based on an emulated environment of the Minos architecture [14] and describe our experiences and observations capturing and analyzing attacks. The main advantage of a Minos-enabled honeypot is that exploits based on corrupting control data can be stopped at the critical point where control flow is hijacked from the legitimate program, facilitating a detailed analysis of the exploit.

Although Minos hardware has not yet been implemented, we are able to deploy Minos systems with the Bochs full system Pentium emulator. We discuss complexities of the exploits Minos has caught that are not accounted for in the simple model of “buffer overflow exploits” prevalent in the literature. We then propose the Epsilon-Gamma-Pi model to describe control data attacks in a way that is useful towards understanding polymorphic techniques. This model can not only aim at the centers of the concepts of exploit vector (ε), bogus control data (γ), and payload (π) but also give them shape. This paper will quantify the polymorphism available to an attacker for γ and π, while so characterizing ε is left for future work.

[17] L. de Moura, S. Owre, H. Rueß, J. Rushby, N. Shankar, M. Sorea, and A. Tiwari. SAL 2. In R. Alur and D. Peled, editors, *Computer-Aided Verification, CAV*, volume 3114 of *LNCS*, pages 496-500. Springer, July 2004. [bib]

[18] D. Ellis. Worm anatomy and model. In *Proceedings of the 2003 ACM Workshop on Rapid Malcode*, pages 42-50. ACM Press, 2003. [bib | DOI]

We present a general framework for reasoning about network worms and analyzing the potency of worms within a specific network. First, we present a discussion of the life cycle of a worm based on a survey of contemporary worms. We build on that life cycle by developing a relational model that associates worm parameters, attributes of the environment, and the subsequent potency of the worm. We then provide a worm analytic framework that captures the generalized mechanical process a worm goes through while moving through a specific environment and its state as it does so. The key contribution of this work is a worm analytic framework. This framework can be used to evaluate worm potency and develop and validate defensive countermeasures and postures in both static and dynamic worm conflict. This framework will be implemented in a modeling and simulation language in order to evaluate the potency of specific worms within an environment.

[19] A. Ganesh, D. Gunawardena, P. Key, L. Massouli, and J. Scott. Efficient quarantining of scanning worms: optimal detection and coordination. In *Proceedings of the IEEE INFOCOM*, April 2006. [bib | .pdf]

[20] G. R. Ganger, G. Economou, and S. M. Bielski. Self-securing network interfaces: What, why and how. Technical report, Computer Science Department, Carnegie Mellon University, 2002. [bib]

Self-securing network interfaces (NIs) examine the packets that they move between network links and host software, looking for and potentially blocking malicious network activity. This paper describes self-securing network interfaces, their features, and examples of how these features allow administrators to more effectively spot and contain malicious network activity. We present a software architecture for self-securing NIs that separates scanning software into applications (called scanners) running on a NI kernel. The resulting scanner API simplifies the construction of scanning software and allows its powers to be contained even if it is subverted. We illustrate the potential via a prototype

self-securing NI and two example scanners: one that identifies and blocks known e-mail viruses and one that identifies and inhibits rapidly-propagating worms like Code-Red.

[21] Know your enemy: Tracking botnets, March 2005. http://www.honeynet.org/papers/bots/. [bib]

[22] S. P. Gorman, R. G. Kulkarni, L. A. Schintler, and R. R. Stough. Least effort strategies for cybersecurity. Technical report, George Mason University, 2003. [bib]

Cybersecurity is an issue of increasing concern since the events of September 11th. Many questions have been raised concerning the security of the Internet and the rest of the US's information infrastructure. This paper begins to examine the issue by analyzing the Internet's autonomous system (AS) map. Using the AS map, malicious infections are simulated and different defense strategies are considered in a cost benefit framework. The results show that protecting the most connected nodes provides significant gains in security and that after the small minority of most connected nodes are protected there are diminishing returns for further protection. Although if parts of the small minority are not protected, such as non-US networks, protection levels are significantly decreased.

[23] M. Gualtieri and D. Mossé. Limiting worms via QoS degradation. Technical report, Computer Science Department, University of Pittsburgh, 2003. [bib]

[24] A. Haeberlen, A. Mislove, A. Post, and P. Druschel. Fallacies in evaluating decentralized systems. In *Proceedings of the 5th International Workshop on Peer-to-Peer Systems (IPTPS)*, February 2006. [bib | .html]

Research on decentralized systems such as peer-to-peer overlays and ad hoc networks has been hampered by the fact that few systems of this type are in production use, and the space of possible applications is still poorly understood. As a consequence, new ideas have mostly been evaluated using common synthetic workloads, traces from a few existing systems, testbeds like PlanetLab, and simulators like ns-2. Some of these methods have, in fact, become the “gold standard” for evaluating new systems, and are often a prerequisite for getting papers accepted at top conferences in the field. In this paper, we examine the current practice of evaluating decentralized systems under these specific sets of conditions and point out pitfalls associated with this practice. In particular, we argue that (i) despite authors' best intentions, results from such evaluations often end up being inappropriately generalized; (ii) there is an incentive not to deviate from the accepted standard of evaluation, even if that is technically appropriate; (iii) research may gravitate towards systems that are feasible and perform well when evaluated in the accepted environments; and, (iv) in the worst-case, research may become ossified as a result. We close with a call to action for the community to develop tools, data, and best practices that allow systems to be evaluated across a space of workloads and environments.

[25] J. Kannan, L. Subramanian, I. Stoica, and R. H. Katz. Analyzing cooperative containment of fast scanning worms. In *Proceedings of USENIX Steps to Reducing Unwanted Traffic on the Internet Workshop (SRUTI)*, pages 17-23, July 2005. [bib | .html | .pdf]

Fast scanning worms, that can infect nearly the entire vulnerable population in order of minutes, are among the most serious threats to the Internet today. In this work, we investigate the efficacy of cooperation among Internet firewalls in containing such worms. We first propose a model for firewall-level cooperation and then study the containment in our model of cooperation using analysis and simulation. Our results suggest that, with moderate overhead, cooperation among Internet firewalls can provide 95% containment under 10% deployment while being resilient to 100-1000 malicious firewalls.

[26] V. Karamcheti, D. Geiger, Z. Kedem, and S. Muthukrishnan. Detecting malicious network traffic using inverse distributions of packet contents. In *MineNet '05: Proceeding of the 2005 ACM SIGCOMM workshop on Mining network data*, pages 165-170, New York, NY, USA, 2005. ACM Press. [bib | DOI]

We study the problem of detecting malicious IP traffic in the network early, by analyzing the contents of packets. Existing systems look at packet contents as a bag of substrings and study characteristics of its base distribution B where B(i) is the frequency of substring i.We propose

studying the inverse distribution I where I(f) is the number of substrings that appear with frequency f. As we show using a detailed case study, the inverse distribution shows the emergence of malicious traffic very clearly not only in its “static” collection of bumps, but also in its nascent “dynamic” state when the phenomenon manifests itself only as a distortion of the inverse distribution envelope. We describe our probabilistic analysis of the inverse distribution in terms of Gaussian mixtures, our preliminary solution for discovering these bumps automatically. Finally, we briefly discuss challenges in analyzing the inverse distribution of IP contents and its applications.

[27] A. D. Keromytis, J. Parekh, P. N. Gross, G. Kaiser, V. Misra, J. Nieh, D. Rubenstein, and S. Stolfo. A holistic approach to service survivability. In *Proceedings of the 2003 ACM Workshop on Survivable and Self-Regenerative Systems (SSRS)*, pages 11-22, 2003. [bib | DOI]

We present SABER (Survivability Architecture: Block, Evade, React), a proposed survivability architecture that blocks, evades and reacts to a variety of attacks by using several security and survivability mechanisms in an automated and coordinated fashion. Contrary to the ad hoc manner in which contemporary survivable systems are built-using isolated, independent security mechanisms such as firewalls, intrusion detection systems and software sandboxes-SABER integrates several different technologies in an attempt to provide a unified framework for responding to the wide range of attacks malicious insiders and outsiders can launch.

This coordinated multi-layer approach will be capable of defending against attacks targeted at various levels of the network stack, such as congestion-based DoS attacks, software-based DoS or code-injection attacks, and others. Our fundamental insight is that while multiple lines of defense are useful, most conventional, uncoordinated approaches fail to exploit the full range of available responses to incidents. By coordinating the response, the ability to survive successful security breaches increases substantially.

We discuss the key components of SABER, how they will be integrated together, and how we can leverage on the promising results of the individual components to improve survivability in a variety of coordinated attack scenarios. SABER is currently in the prototyping stages, with several interesting open research topics.

[28] H.-A. Kim and B. Karp. Autograph: Toward automated, distributed worm signature detection. In *USENIX Security Symposium*, pages 271-286, 2004. [bib | .html]

Today's Internet intrusion detection systems (IDSes) monitor edge networks' DMZs to identify and/or filter malicious flows. While an IDS helps protect the hosts on its local edge network from compromise and denial of service, it cannot alone effectively intervene to halt and reverse the spreading of novel Internet worms. Generation of the *worm signatures* required by an IDS-the byte patterns sought in monitored traffic to identify worms-today entails non-trivial human labor, and thus significant delay: as network operators detect anomalous behavior, they communicate with one another and manually study packet traces to produce a worm signature. Yet intervention must occur early in an epidemic to halt a worm's spread. In this paper, we describe Autograph, a system that *automatically* generates signatures for novel Internet worms that propagate using TCP transport. Autograph generates signatures by analyzing the prevalence of *portions of flow payloads*, and thus uses no knowledge of protocol semantics above the TCP level. It is designed to produce signatures that exhibit high *sensitivity* (high true positives) and high *specificity* (low false positives); our evaluation of the system on real DMZ traces validates that it achieves these goals. We extend Autograph to share port scan reports among distributed monitor instances, and using trace-driven simulation, demonstrate the value of this technique in speeding the generation of signatures for novel worms. Our results elucidate the fundamental trade-off between early generation of signatures for novel worms and the specificity of these generated signatures.

[29] C. Kreibich and J. Crowcroft. Honeycomb: creating intrusion detection signatures using honeypots. *SIGCOMM Comput. Commun. Rev.*, 34(1):51-56, 2004. [bib | DOI]

This paper describes a system for automated generation of attack signatures for network intrusion detection systems. Our system applies pattern-matching techniques and protocol conformance checks on multiple levels in the protocol hierarchy to network traffic captured a honeypot system. We present results of running the system on an unprotected cable modem connection for 24 hours. The

system successfully created precise traffic signatures that otherwise would have required the skills and time of a security officer to inspect the traffic manually.

[30] Z. Liang and R. Sekar. Automatic generation of buffer overflow attack signatures: An approach based on program behavior models. In *Proceedings of the 21st Annual Computer Security Applications Conference (ACSAC)*, pages 215-224, 2005. [bib | DOI]

Buffer overflows have become the most common target for network-based attacks. They are also the primary mechanism used by worms and other forms of automated attacks. Although many techniques have been developed to prevent server compromises due to buffer overflows, these defenses still lead to server crashes. When attacks occur repeatedly, as is common with automated attacks, these protection mechanisms lead to repeated restarts of the victim application, rendering its service unavailable. To overcome this problem, we develop a new approach that can learn the characteristics of a particular attack, and filter out future instances of the same attack or its variants. By doing so, our approach significantly increases the availability of servers subjected to repeated attacks. The approach is fully automatic, does not require source code, and has low runtime overheads. In our experiments, it was effective against most attacks, and did not produce any false positives.

[31] Z. Liang and R. Sekar. Fast and automated generation of attack signatures: a basis for building self-protecting servers. In *Proceedings of the 12th ACM Conference on Computer and Communications Security (CCS)*, pages 213-222, November 2005. [bib | DOI]

Large-scale attacks, such as those launched by worms and zombie farms, pose a serious threat to our network-centric society. Existing approaches such as software patches are simply unable to cope with the volume and speed with which new vulnerabilities are being discovered. In this paper, we develop a new approach that can provide effective protection against a vast majority of these attacks that exploit memory errors in C/C++ programs. Our approach, called COVERS, uses a forensic analysis of a victim server's memory to correlate attacks to inputs received over the network, and *automatically* develop a signature that characterizes inputs that carry attacks. The signatures tend to capture characteristics of the underlying vulnerability (e.g., a message field being too long) rather than the characteristics of an attack, which makes them effective against variants of attacks. Our approach introduces low overheads (under 10%), does not require access to source code of the protected server, and has successfully generated signatures for the attacks studied in our experiments, without producing false positives. Since the signatures are generated in tens of milliseconds, they can potentially be distributed quickly over the Internet to filter out (and thus stop) fast-spreading worms. Another interesting aspect of our approach is that it can defeat guessing attacks reported against address-space randomization and instruction set randomization techniques. Finally, it increases the capacity of servers to withstand repeated attacks by a factor of 10 or more.

[32] M. Liljenstam and D. M. Nicol. Comparing passive and active worm defenses. In *Proceedings of the First International Conference on the Quantitative Evaluation of Systems (QEST)*, pages 18-27, September 2004. [bib | .pdf]

[33] M. Liljenstam, D. M. Nicol, V. H. Berk, and R. S. Gray. Simulating realistic network worm traffic for worm warning system design and testing. In *Proceedings of the 2003 ACM Workshop on Rapid Malcode (WORM)*, pages 24-33. ACM Press, 2003. [bib | DOI]

Reproducing the effects of large-scale worm attacks in a laboratory setup in a realistic and reproducible manner is an important issue for the development of worm detection and defense systems. In this paper, we describe a worm simulation model we are developing to accurately model the large-scale spread dynamics of a worm and many aspects of its detailed effects on the network. We can model slow or fast worms with realistic scan rates on realistic IP address spaces and selectively model local detailed network behavior. We show how it can be used to generate realistic input traffic for a working prototype worm detection and tracking system, the Dartmouth ICMP BCC: System/Tracking and Fusion Engine (DIB:S/TRAFEN), allowing performance evaluation of the system under realistic conditions. Thus, we can answer important design questions relating to necessary detector coverage and noise filtering without deploying and operating a full system. Our experiments indicate that the tracking algorithms currently implemented in the DIB:S/TRAFEN system could detect attacks such as Code Red v2 and Sapphire/Slammer very early, even when monitoring a quite limited portion of the address space, but more sophisticated algorithms are being constructed to reduce the risk of false positives in the presence of significant “background noise” scanning.

[34] M. Liljenstam, Y. Yuan, B. J. Premore, and D. M. Nicol. A mixed abstraction level simulation model of large-scale Internet worm infestations. In *10th International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 109-116. IEEE Computer Society, 2002. [bib | http]

Large-scale worm infestations, such as last year's Code Red, Code Red II, and Nimda, have led to increased interest in modeling these events to assess threat levels, evaluate countermeasures and investigate possible influence on the Internet infrastructure. However, the inherently large scale of these phenomena pose significant challenges for models that include infrastructure detail. We explore the use of selective abstraction through epidemiological models in conjunction with detailed protocol models as a means to scale up simulations to a point where we can ask meaningful questions regarding a hypothesized link between worms and inter-domain routing instability. We find that this approach shows significant promise, in contrast to some of our early attempts using all-out packet level models. We also describe some approaches we are taking to collect the underlying data for our models.

[35] P. Liu, W. Zang, and M. Yu. Incentive-based modeling and inference of attacker intent, objectives, and strategies. *ACM Trans. Inf. Syst. Secur.*, 8(1):78-118, 2005. [bib | DOI]

[36] M. Locasto, J. Parekh, A. Keromytis, and S. Stolfo. Towards collaborative security and P2P intrusion detection. In *Proceedings from the Sixth Annual IEEE Systems, Man and Cybernetics (SMC) Information Assurance Workshop*, pages 333-339, June 2005. [bib | DOI]

The increasing array of Internet-scale threats is a pressing problem for every organization that utilizes the network. Organizations have limited resources to detect and respond to these threats. The end-to-end (E2E) sharing of information related to probes and attacks is a facet of an emerging trend toward “collaborative security”. The key benefit of a collaborative approach to intrusion detection is a better view of global network attack activity. Augmenting the information obtained at a single site with information gathered from across the network can provide a more precise model of an attacker's behavior and intent. While many organizations see value in adopting such a collaborative approach, some challenges must be addressed before intrusion detection can be performed on an inter-organizational scale. We report on our experience developing and deploying a decentralized system for efficiently distributing alerts to collaborating peers. Our system, worminator, extracts relevant information from alert streams and encodes it in bloom filters. This information forms the basis of a distributed watchlist. The watchlist can be distributed via a choice of mechanisms ranging from a centralized trusted third party to a decentralized P2P-style overlay network.

[37] P. McDaniel, S. Sen, O. Spatscheck, J. V. der Merwe Bill Aiello, and C. Kalmanek. Enterprise security: A community of interest based approach. In *Proceedings of Network and Distributed Systems Security (NDSS)*, February 2006. (draft). [bib | .pdf]

Enterprise networks today carry a range of mission critical communications. A successful worm attack within an enterprise network can be substantially more devastating to most companies than attacks on the larger Internet. In this paper we explore a brownfield approach to hardening an enterprise network against active malware such as worms. The premise of our approach is that if future communication patterns are constrained to historical “normal” communication patterns, then the ability of malware to exploit vulnerabilities in the enterprise can be severely curtailed. We present techniques for automatically deriving individual host profiles that capture historical communication patterns (i.e., community of interest (COI)) of end hosts within an enterprise network. Using traces from a large enterprise network, we investigate how a range of different security policies based on these profiles impact usability (as valid communications may get restricted) and security (how well the policies contain malware). Our evaluations indicate that a simple security policy comprising our *Extended COI-based profile and relaxed Throttling Discipline* can effectively contains worm behavior within an enterprise without significantly impairing normal network operation.

[38] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. Inside the Slammer worm. *IEEE Security and Privacy*, 1(4):33-39, July-August 2003. [bib | DOI]

The Slammer worm, also sometimes known as Sapphire, was the fastest worm in history, achieving a

peak scanning rate of 55 million scans per second. This time, this new worm breed had internal programming flaws and a benign payload, but what about next time?

[39] D. Moore, C. Shannon, G. Voelker, and S. Savage. Internet quarantine: Requirements for containing self-propagating code. In *Proceedings of the 2003 IEEE Infocom Conference (INFOCOM)*, April 2003. [bib | .pdf]

[40] J. Newsome, B. Karp, and D. X. Song. Polygraph: Automatically generating signatures for polymorphic worms. In *IEEE Symposium on Security and Privacy*, pages 226-241, May 2005. [bib | DOI]

It is widely believed that content-signature-based intrusion detection systems (IDS) are easily evaded by polymorphic worms, which vary their payload on every infection attempt. In this paper, we present Polygraph, a signature generation system that successfully produces signatures that match polymorphic worms. Polygraph generates signatures that consist of multiple disjoint content substrings. In doing so, Polygraph leverages our insight that for a real-world exploit to function properly, multiple invariant substrings must often be present in all variants of a payload; these substrings typically correspond to protocol framing, return addresses, and in some cases, poorly obfuscated code. We contribute a definition of the polymorphic signature generation problem; propose classes of signature suited for matching polymorphic worm payloads; and present algorithms for automatic generation of signatures in these classes. Our evaluation of these algorithms on a range of polymorphic worms demonstrates that Polygraph produces signatures for polymorphic worms that exhibit low false negatives and false positives.

[41] D. Nicol and M. Liljenstam. Models of active worm defenses. In *Proceedings of the Measurement, Modeling and Analysis of the Internet Workshop (IMA)*, January 2004. [bib | .pdf]

[42] D. Nojiri, J. Rowe, and K. Levitt. Cooperative response strategies for large scale attack mitigation. In *DARPA Information Survivability Conference and Exposition*, pages 293-302, 2003. [bib]

[43] A. Pnueli. The temporal logic of programs. In *Proceedings of the 18th IEEE Symposium on Foundations of Computer Science*, pages 46-67, 1977. [bib]

[44] P. Porras, L. Briesemeister, K. Skinner, K. Levitt, J. Rowe, and Y.-C. A. Ting. A hybrid quarantine defense. In *Proceedings of the 2004 ACM workshop on Rapid malcode (WORM)*, pages 73-82. ACM Press, 2004. [bib | DOI | .pdf]

We study the strengths, weaknesses, and potential synergies of two complementary worm quarantine defense strategies under various worm attack profiles. We observe their abilities to delay or suppress infection growth rates under two propagation techniques and three scan rates, and explore the potential synergies in combining these two complementary quarantine strategies. We compare the performance of the individual strategies against a hybrid combination strategy, and conclude that the hybrid strategy yields substantial performance improvements, beyond what either technique provides independently. This result offers potential new directions in hybrid quarantine defenses.

[45] G. Portokalidis and H. Bos. SweetBait: Zero-hour worm detection and containment using honeypots. Technical Report IR-CS-015, Vrije Universiteit Amsterdam, May 2005. [bib | .pdf]

[46] N. Provos. A virtual honeypot framework. In *Proceedings of the 12th USENIX Security Symposium*, pages 1-14, August 2004. [bib]

[47] S. Qing and W. Wen. A survey and trends on Internet worms. *Computers & Security*, 24(4):334-346, June 2005. [bib | DOI]

With the explosive growth and increasing complexity of network applications, the threats of Internet worms against network security are more and more serious. This paper presents the concepts and research situations of Internet worms, their function component, and their execution mechanism. It also addresses the scanning strategies, propagation models, and the critical techniques of Internet worm prevention. Finally, the remaining problems and emerging trends in this area are also outlined.

[48] G. F. Riley, M. I. Sharif, and W. Lee. Simulating internet worms. In *Proceedings of the 12th International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 268-274, 2004. [bib | .pdf]

The accurate and efficient modeling of Internet worms is a particularly challenging task for network simulation tools. The atypical and aggressive behavior of these worms can easily consume excessive resources, both processing time and storage, within a typical simulator. In particular, the selection of random IP addresses, and the sending of packets to the selected hosts, even if they are non-existent or not modeled in the simulation scenario, is challenging for existing network simulation tools. Further, the computation of routing information for these randomly chosen target addresses defeats most caching or on-demand routing methods, resulting in substantial overhead in the simulator. We discuss the design of our Internet worm models in the Georgia Tech Network Simulator, and show how we addressed these issues.We present some results from our Internet worm simulations that show the rate of infection spread for a typical worm under a variety of conditions.

[49] The SAL intermediate language, 2003. Computer Science Laboratory, SRI International, Menlo Park, CA. http://sal.csl.sri.com/. [bib]

[50] R. Scandariato and J. Knight. The design and evaluation of a defense system for Internet worms. In *Proceedings of the 23rd IEEE International Symposium on Reliable Distributed Systems (SRDS)*, pages 164-173, October 2004. [bib | DOI]

Many areas of society have become heavily dependent on services such as transportation facilities, utilities and so on that are implemented in part by large numbers of computers and communications links. Both past incidents and research studies show that a well-engineered Internet worm can disable such systems in a fairly simple way and, most notably, in a matter of a few minutes. This indicates the need for defenses against worms but their speed rules out the possibility of manually countering worm outbreaks. We present a platform that emulates the epidemic behavior of Internet active worms in very large networks. A reactive control system operates on top of the platform and provides a monitor/analyze/respond approach to deal with infections automatically. Details of our highly configurable platform and various experimental performance results are presented.

[51] S. Sellke, N. B. Shroff, and S. Bagchi. Modeling and automated containment of worms. In *Proceedings of the International Conference on Dependable Systems and Networks (DSN)*, pages 528-537, 2005. [bib | DOI | .pdf]

Self-propagating codes, called worms, such as Code Red, Nimda, and Slammer, have drawn significant attention due to their enormous adverse impact on the Internet. There is a great interest in the research community in modeling the spread of worms and in providing adequate defense mechanisms against them. In this paper, we present a (stochastic) branching process model for characterizing the propagation of Internet worms. This model leads to the development of an automatic worm containment strategy that prevents the spread of worms beyond its early stages. Specifically, using the branching process model, we are able to (1) provide a precise condition that determines whether the worm will eventually die out and (2) provdide the probability that the total number of hosts that the worm infects will be below a certain level. We use these insights to develop a simple automatic worm containment scheme, which is demonstrated, through simulations and real trace data, to be both effective and non-intrusive.

[52] J. F. Shoch and J. A. Hupp. The “worm” programs-early experience with a distributed computation. *Communications of the ACM*, 25(3):172-180, 1982. [bib | DOI]

The “worm” programs were an experiment in the development of distributed computations: programs that span machine boundaries and also replicate themselves in idle machines. A “worm” is composed of multiple “segments,” each running on a different machine. The underlying worm maintenance mechanisms are responsible for maintaining the worm-finding free machines when needed and replicating the program for each additional segment. These techniques were successfully used to support several real applications, ranging from a simple multimachine test program to a more sophisticated real-time animation system harnessing multiple machines.

[53] K. Simkhada, H. Tsunoda, Y. Waizumi, and Y. Nemoto. Differencing worm flows and normal

flows for automatic generation of worm signatures. In *Proceedings of the Seventh IEEE International Symposium on Multimedia (ISM)*, pages 680-685, December 2005. [bib | DOI]

Internet worms pose a serious threat to networks. Most current Intrusion Detection Systems (IDSs) take signature matching approach to detect worms. Given the fact that most signatures are developed manually, generating new signatures for each variant of a worm incurs significant overhead. In this paper, we propose a difference-based scheme which differences worm flows and normal flows to generate robust worm signatures. The proposed scheme is based on two observational facts - worm flows contain several invariant portions in their payloads, and core worm codes do not exist in normal flows. It uses samples of worm flows detected by available means to extract common tokens. It then differences the set of these tokens with those of normal flows and generates signature candidates. By using such signatures within enterprises, out of reach of worm writers, the possibility of being tricked by worm writers can be reduced. We evaluate the proposed scheme using real network traffic traces that contains worms. Experiment results show that the proposed scheme exhibits high detection rate with low false positives.

[54] S. Singh, C. Estan, G. Varghese, and S. Savage. The EarlyBird system for realtime detection of unknown worms. Technical Report CS2003-0761, UC San Diego, August 2003. [bib]

Network worms are a major threat to the security of today's Internet-connected hosts and networks. The combination of unmitigated connectivity and widespread software homogeneity allows worms to exploit tremendous parallelism in propagation. Modern worms spread so quickly that no human-mediated reaction to the outbreak of a new worm can hope to prevent a widespread epidemic. In this paper we propose an automated method for detecting new worms based on traffic characteristics common to most of them: highly repetitive packet content, an increasing population of sources generating infections and an increasing number of destinations being targeted. Our method generates content signatures for the worm without any human intervention. Preliminary results on a small network show promising results: we have identified three confirmed worms with a low percentage of false positives. This gives us reason to believe that our method could form the core of an effective network-level worm detection and countermeasure system capable of substantially slowing down the spread of new worms.

[55] S. Singh, C. Estan, G. Varghese, and S. Savage. Automated worm fingerprinting. In *Proceedings of the 6th Symposium on Operating Systems Design and Implementation (OSDI)*, pages 45-60, December 2004. [bib | .html | .pdf]

Network worms are a clear and growing threat to the security of today's Internet-connected hosts and networks. The combination of the Internet's unrestricted connectivity and widespread software homogeneity allows network pathogens to exploit tremendous parallelism in their propagation. In fact, modern worms can spread so quickly, and so widely, that no human-mediated reaction can hope to contain an outbreak.

In this paper, we propose an automated approach for quickly detecting previously unknown worms and viruses based on two key behavioral characteristics-a common exploit sequence together with a range of unique sources generating infections and destinations being targeted. More importantly, our approach-called “content sifting”-automatically generates *precise* signatures that can then be used to filter or moderate the spread of the worm *elsewhere* in the network.

Using a combination of existing and novel algorithms we have developed a scalable content sifting implementation with low memory and CPU requirements. Over months of active use at UCSD, our *Earlybird* prototype system has automatically detected and generated signatures for all pathogens known to be active on our network as well as for several *new* worms and viruses which were *unknown* at the time our system identified them. Our initial experience suggests that, for a wide range of network pathogens, it may be practical to construct fully automated defenses-even against so-called “zero-day” epidemics.

[56] Scalable Simulation Framework. http://www.ssfnet.org/. [bib]

[57] S. Staniford. Containment of scanning worms in enterprise networks. *Journal of Computer Security*, to appear. [bib]

[58] S. Staniford, V. Paxson, and N. Weaver. How to 0wn the Internet in your spare time. In

Proceedings of the 11th USENIX Security Symposium, Aug. 2002. [bib]

> The ability of attackers to rapidly gain control of vast numbers of Internet hosts poses an immense risk to the overall security of the Internet. Once subverted, these hosts can not only be used to launch massive denial of service floods, but also to steal or corrupt great quantities of sensitive information, and confuse and disrupt use of the network in more subtle ways.
>
> We present an analysis of the magnitude of the threat. We begin with a mathematical model derived from empirical data of the spread of Code Red I in July, 2001. We discuss techniques subsequently employed for achieving greater virulence by Code Red II and Nimda. In this context, we develop and evaluate several new, highly virulent possible techniques: hit-list scanning (which creates a *Warhol* worm), permutation scanning (which enables self-coordinating scanning), and use of Internet-sized hit-lists (which creates a *flash worm*).
>
> We then turn to the to the threat of *surreptitious* worms that spread more slowly but in a much harder to detect “contagion” fashion. We demonstrate that such a worm today could arguably subvert upwards of 10,000,000 Internet hosts. We also consider robust mechanisms by which attackers can control and update deployed worms.
>
> In conclusion, we argue for the pressing need to develop a “Center for Disease Control” analog for virus- and worm-based threats to national cybersecurity, and sketch some of the components that would go into such a Center.

[59] S. Staniford-Chen et al. GrIDS-A graph based intrusion detection system for large networks. In *Proceedings of the 19th National Information Systems Security Conference*, volume 1, pages 361-370, October 1996. [bib]

[60] P. Stephenson. Modeling a virus or worm attack. *Computer Fraud & Security*, (9):15-19, September 2004. [bib | DOI]

> In our last two columns we introduced the concepts of modeling and simulation, security policy domains and the use of Colored Petri Nets (CPNets). In this column we will take the Net that we created in our last column (and discussed very briefly) and describe in more detail how we built it and how we use it to simulate security-relevant network activity. We begin by reviewing, briefly, the CPNet we created in the last column.

[61] S. Stolfo. *The Black Book on Corporate Security*, chapter Collaborative Security: Uniting Against a Common Foe, pages 219-237. 2005. [bib]

[62] P. Szor and P. Ferrie. Hunting for metamorphic. Symantec Security Response. White Paper., June 2003. [bib | http]

[63] H. Toyoizumi and A. Kara. Predators: Good will mobile codes combat against computer viruses. In *Proceedings of the 2002 Workshop on New Security Paradigms (NSPW)*, pages 11-17, 2002. [bib | DOI]

> We present a mathematical analysis of a new approach to fight against computer viruses through the use of their predators. Predators are good will mobile codes which, like viruses, travel over computer networks, and replicate and multipy themselves. The only difference is that predators are specifically designed to eliminate the viruses. We model the interaction between predators and viruses by the Lotka-Volterra equations, which are widely used in mathematical biology. Using this model, we derive a method to constrain the number of predators to be as few as possible, while maintaining their power to eliminate viruses.

[64] G. van't Noordende, F. Brazier, and A. Tanenbaum. Security in a mobile agent system. In *First IEEE Symposium on Multi-Agent Security and Survivability (MAS&S)*, pages 35-45, Aug 2004. [bib | .pdf]

[65] H. Wang, C. Guo, D. Simon, and A. Zugenmaier. Shield: Vulnerability-driven network filters for preventing known vulnerability exploits. Technical report, Microsoft Research, Technical Report MSR-TR-2003-81, 2004. [bib]

[66] H. J. Wang, C. Guo, D. R. Simon, and A. Zugenmaier. Shield: vulnerability-driven network filters for preventing known vulnerability exploits. In *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, pages 193-204. ACM Press, 2004. [bib | DOI]

Software patching has not been effective as a first-line defense against large-scale worm attacks, even when patches have long been available for their corresponding vulnerabilities. Generally, people have been reluctant to patch their systems immediately, because patches are perceived to be unreliable and disruptive to apply. To address this problem, we propose a first-line worm defense in the network stack, using shields - vulnerability-specific, exploit-generic network filters installed in end systems once a vulnerability is discovered, but before a patch is applied. These filters examine the incoming or outgoing traffic of vulnerable applications, and correct traffic that exploits vulnerabilities. Shields are less disruptive to install and uninstall, easier to test for bad side effects, and hence more reliable than traditional software patches. Further, shields are resilient to polymorphic or metamorphic variations of exploits [62]. In this paper, we show that this concept is feasible by describing a prototype Shield framework implementation that filters traffic above the transport layer. We have designed a safe and restrictive language to describe vulnerabilities as partial state machines of the vulnerable application. The expressiveness of the language has been verified by encoding the signatures of several known vulnerabilites. Our evaluation provides evidence of Shield's low false positive rate and small impact on application throughput. An examination of a sample set of known vulnerabilities suggests that Shield could be used to prevent exploitation of a substantial fraction of the most dangerous ones.

[67] X. Wang and M. K. Reiter. Defending against denial-of-service attacks with puzzle auctions. In *SP '03: Proceedings of the 2003 IEEE Symposium on Security and Privacy*, page 78, Washington, DC, USA, 2003. IEEE Computer Society. [bib | DOI]

Although client puzzles represent a promising approach to defend against certain classes of denial-of-service attacks, several questions stand in the way of their deployment in practice: e.g., how to set the puzzle difficulty in the presence of an adversary with unknown computing power, and how to integrate the approach with existing mechanisms. In this paper, we attempt to address these questions with a new puzzle mechanism called the puzzle auction. Our mechanism enables each client to “bid” for resources by tuning the difficulty of the puzzles it solves, and to adapt its bidding strategy in response to apparent attacks. We analyze the effectiveness of our auction mechanism and further demonstrate it using an implementation within the TCP protocol stack of the Linux kernel. Our implementation has several appealing properties. It effectively defends against SYN ooding attacks, is fully compatible with TCP, and even provides a degree of interoperability with clients with unmodified kernels: Even without a puzzle-solving kernel, a client still can connect to a puzzle auction server under attack (albeit less effectively than those with puzzle-solving kernels, and at the cost of additional server expense).

[68] Y. Wang and C. Wang. Modeling the effects of timing parameters on virus propagation. In *Proceedings of the 2003 ACM Workshop on Rapid Malcode (WORM)*, pages 61-66, 2003. [bib | DOI]

In this paper, we investigate epidemiological models to reason about computer viral propagation. We extend the classical homogeneous models to incorporate two timing parameters: Infection delay and user vigilance. We show that these timing parameters greatly influence the propagation of viral epidemics, and that the explicit treatment of these parameters gives rise to a more realistic and accurate propagation model. We validate the new model with simulation analysis.

[69] N. Weaver, I. Hamadeh, G. Kesidis, and V. Paxson. Preliminary results using scale-down to explore worm dynamics. In *Proceedings of the 2004 ACM Workshop on Rapid Malcode (WORM)*, pages 65-72, 2004. [bib | DOI]

A major challenge when attempting to analyze and model large-scale Internet phenomena such as the dynamics of global worm propagation is finding appropriate abstractions that allow us to tractably grapple with size of the artifact while still capturing its most salient properties. We present initial results from investigating “scaledown” techniques for approximating global Internet worm dynamics by shrinking the effective size of the network under study. We explore scaledown in the context of both

simulation and analysis, using as a calibration touchstone an attempt to reproduce the empirically observed behavior of the Slammer worm, which exhibited a peculiar decline in average per-worm scanning rate not seen in other worms (except for the later Witty worm, which exhibited similar propagation dynamics). We develop a series of abstract models approximating Slammer's Internet propagation and demonstrate that such modeling appears to require incorporating both heterogeneous clustering of infectibles and heterogeneous access-link bandwidths connecting those clusters to the Internet core. We demonstrate the viability of scaledown but also explore two important artifacts it introduces: heightened variability of results, and biasing the worm towards earlier propagation.

[70] N. Weaver, V. Paxson, S. Staniford, and R. Cunningham. A taxonomy of computer worms. In *Proceedings of the 2003 ACM Workshop on Rapid Malcode (WORM)*, pages 11-18, 2003. [bib | DOI]

To understand the threat posed by computer worms, it is necessary to understand the classes of worms, the attackers who may employ them, and the potential payloads. This paper describes a preliminary taxonomy based on worm target discovery and selection strategies, worm carrier mechanisms, worm activation, possible payloads, and plausible attackers who would employ a worm.

[71] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar. An integrated experimental environment for distributed systems and networks. In *Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI)*, pages 255-270. USENIX Association, December 2002. [bib]

Three experimental environments traditionally support network and distributed systems research: network emulators, network simulators, and live networks. The continued use of multiple approaches highlights both the value and inadequacy of each. Netbed, a descendant of Emulab, provides an experimentation facility that integrates these approaches, allowing researchers to configure and access networks composed of emulated, simulated, and wide-area nodes and links. Netbed's primary goals are ease of use, control, and realism, achieved through consistent use of virtualization and abstraction.

By providing operating system-like services, such as resource allocation and scheduling, and by virtualizing heterogeneous resources, Netbed acts as a virtual machine for network experimentation. This paper presents Netbed's overall design and implementation and demonstrates its ability to improve experimental automation and efficiency. These, in turn, lead to new methods of experimentation, including automated parameter-space studies within emulation and straightforward comparisons of simulated, emulated, and wide-area scenarios.

[72] D. Whyte, E. Kranakis, and P. van Oorschot. DNS-based detection of scanning worms in an enterprise network. In *Proceedings of the 12th Network and Distributed System Security Symposium (NDSS)*, pages 181-195, February 2005. [bib]

[73] M. M. Williamson. Throttling viruses: Restricting propagation to defeat malicious mobile code. In *Proceedings of the 18th Annual Computer Security Applications Conference*, page 61. IEEE Computer Society, 2002. [bib | DOI]

Modern computer viruses spread incredibly quickly, far faster than human-mediated responses. This greatly increases the damage that they cause. This paper presents an approach to restricting this high speed propagation automatically. The approach is based on the observation that during virus propagation, an infected machine will connect to as many different machines as fast as possible. An uninfected machine has a different behaviour: connections are made at a lower rate, and are locally correlated (repeat connections to recently accessed machines are likely). This paper describes a simple technique to limit the rate of connections to “new” machines that is remarkably effective at both slowing and halting virus propagation without affecting normal traffic. Results of applying the filter to web browsing data are included. The paper concludes by suggesting an implementation and discussing the potential and limitations of this approach.

[74] C. Wong, C. Wang, D. Song, S. Bielski, and G. R. Ganger. Dynamic quarantine of Internet worms. In *Proceedings of the International Conference on Dependable Systems and Networks*

DSN-2004, June 2004. [bib]

[75] Worminator web site http://worminator.cs.columbia.edu. [bib]

[76] J. Xu and W. Lee. Sustaining availability of web services under severe denial of service attacks. *IEEE Transaction on Computers, special issue on Reliable Distributed Systems*, 52(2):195-208, Feb. 2003. [bib | DOI]

The recent tide of Distributed Denial of Service (DDoS) attacks against high-profile web sites demonstrate how devastating DDoS attacks are and how defenseless the Internet is under such attacks. We design a practical DDoS defense system that can protect the availability of web services during severe DDoS attacks. The basic idea behind our system is to isolate and protect legitimate traffic from a huge volume of DDoS traffic when an attack occurs. Traffic that needs to be protected can be recognized and protected using efficient cryptographic techniques. Therefore, by provisioning adequate resource (e.g., bandwidth) to legitimate traffic separated by this process, we are able to provide adequate service to a large percentage of clients during DDoS attacks. The worst-case performance (effectiveness) of the system is evaluated based on a novel game theoretical framework, which characterizes the natural adversarial relationship between a DDoS adversary and the proposed system. We also conduct a simulation study to verify a key assumption used in the game-theoretical analysis and to demonstrate the system dynamics during an attack.

[77] V. Yegneswaran, P. Barford, and D. Plonka. On the design and use of Internet sinks for network abuse monitoring. In *Proceedings of the 7th International Symposium Recent Advances in Intrusion Detection (RAID)*, volume 3224 of *Lecture Notes in Computer Science*, pages 146-165, January 2004. [bib | DOI]

Monitoring *unused* or *dark* IP addresses offers opportunities to significantly improve and expand knowledge of abuse activity without many of the problems associated with typical network intrusion detection and firewall systems. In this paper, we address the problem of designing and deploying a system for monitoring large unused address spaces such as class A telescopes with 16M IP addresses. We describe the architecture and implementation of the Internet Sink (iSink) system which measures packet traffic on unused IP addresses in an efficient, extensible and scalable fashion. In contrast to traditional intrusion detection systems or firewalls, iSink includes an *active* component that generates response packets to incoming traffic. This gives the iSink an important advantage in discriminating between different types of attacks (through examination of the response payloads). The key feature of iSink's design that distinguishes it from other unused address space monitors is that its active response component is *stateless* and thus highly scalable. We report performance results of our iSink implementation in both controlled laboratory experiments and from a case study of a live deployment. Our results demonstrate the efficiency and scalability of our implementation as well as the important perspective on abuse activity that is afforded by its use.

Keywords: Intrusion Detection, Honeypots, Deception Systems

[78] Y.-K. Zhang, F.-W. Wang, Y.-Q. Zhang, and J.-F. Ma. Worm propagation modeling and analysis based on quarantine. In *Proceedings of the 3rd International Conference on Information Security (InfoSecu)*, pages 69-75, 2004. [bib | DOI]

In recent years, the worms that had a dramatic increase in the frequency and virulence of such outbreaks have become one of the major threats to the security of the Internet. In this paper, we provide a worm propagating model. It bases on the classical epidemic Kermack-Kermack model, adopts dynamic quarantine strategy, dynamic infecting rate and removing rate. The analysis shows that model can efficiently reduce a worm's propagation speed, which can give us more precious time to defend it, and reduce the negative influence of worms. The simulation results verify the effectiveness of the model.

[79] G. Zhu and J. Dai. Economic perspective of information security. In *Security and Management*, pages 527-533, 2003. [bib]

[80] C. C. Zou, W. Gong, and D. Towsley. Code red worm propagation modeling and analysis. In *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS)*, pages 138-147, 2002. [bib | DOI]

The Code Red worm incident of July 2001 has stimulated activities to model and analyze Internet worm propagation. In this paper we provide a careful analysis of Code Red propagation by accounting for two factors: one is the dynamic countermeasures taken by ISPs and users; the other is the slowed down worm infection rate because Code Red rampant propagation caused congestion and troubles to some routers. Based on the classical epidemic Kermack-Mckendrick model, we derive a general Internet worm model called the two-factor worm model. Simulations and numerical solutions of the two-factor worm model match the observed data of Code Red worm better than previous models do. This model leads to a better understanding and prediction of the scale and speed of Internet worm spreading.

[81] C. C. Zou, D. Towsley, and W. Gong. A firewall network system for worm defense in enterprise networks. Technical Report TR-04-CSE-01, University of Massachusetts Amherst, College of Engineering, February 2004. [bib | .pdf]

[82] C. C. Zou, D. Towsley, and W. Gong. On the performance of Internet worm scanning strategies. *Performance Evaluation*, 2005. In Press, Corrected Proof. [bib | DOI]

In recent years, fast spreading worms, such as Code Red, Slammer, Blaster and Sasser, have become one of the major threats to the security of the Internet. In order to defend against future worms, it is important to first understand how worms propagate and how different scanning strategies affect worm propagation dynamics. In this paper, we systematically model and analyze worm propagation under various scanning strategies, such as uniform scan, routing scan, hit-list scan, cooperative scan, local preference scan, sequential scan, divide-and-conquer scan, target scan, etc. We also provide an analytical model to accurately model Witty worm's destructive behavior. By using the same modeling framework, we reveal the underlying similarity and relationship between different worm scanning strategies. In addition, based on our simulation and analysis of Blaster worm propagation and monitoring, we provide a guideline for building a better worm monitoring infrastructure.
