Internet Quarantine: Requirements for Containing Self-Propagating Code

David Moore, Colleen Shannon, Geoffrey M. Voelker, Stefan Savage
University of California, San Diego

Abstract— **It has been clear since 1988 that self-propagating code can quickly spread across a network by exploiting homogeneous security vulnerabilities. However, the last few years have seen a dramatic increase in the frequency and virulence of such "worm" outbreaks. For example, the Code-Red worm epidemics of 2001 infected hundreds of thousands of Internet hosts in a very short period – incurring enormous operational expense to track down, contain, and repair each infected machine. In response to this threat, there is considerable effort focused on developing technical means for detecting and containing worm infections before they can cause such damage.**

This paper does not propose a particular technology to address this problem, but instead focuses on a more basic question: *How well will any such approach contain a worm epidemic on the Internet?* We describe the design space of worm containment systems using three key parameters – reaction time, containment strategy and deployment scenario. Using a combination of analytic modeling and simulation, we describe how each of these design factors impacts the dynamics of a worm epidemic and, conversely, the minimum engineering requirements necessary to contain the spread of a given worm. While our analysis cannot provide definitive guidance for engineering defenses against all future threats, we demonstrate the lower bounds that any such system must exceed to be useful today. Unfortunately, our results suggest that there are significant technological and administrative gaps to be bridged before an effective defense can be provided in today's Internet.

I. Introduction

On July 19th, 2001, a self-propagating program, or *worm*, was released into the Internet. The worm, dubbed "Code-Red v2", probed random Internet hosts for a documented vulnerability in the popular Microsoft IIS Web server. As susceptible hosts were infected with the worm, they too attempted to subvert other hosts – dramatically increasing the incidence of the infection. Over fourteen hours, the worm infected almost 360,000 hosts, reaching an incidence of 2,000 hosts per minute before peaking [1]. The direct costs of recovering from this epidemic (including subsequent strains of Code-Red) have been estimated in excess of $2.6 billion [2]. While Code-Red was neither the first nor the last widespread computer epidemic, it exemplifies the vulnerabilities present in today's Internet environment. A relatively homogeneous software base coupled with high-bandwidth connectivity provides an ideal climate for self-propagating attacks.

Unfortunately, as demonstrated by the Code-Red episode, we do not currently have an effective defense against such threats. While research in this field is nascent, traditional epidemiology suggests that the most important factors determining the spread of an infectious pathogen are the vulnerability of the population, the length of the infectious period and the rate of infection. These translate into three potential interventions to mitigate the threat of worms: *prevention*, *treatment*, and *containment*. This paper focuses exclusively on the last approach, but we briefly discuss each to justify that decision.

A. Prevention

Prevention technologies are those that reduce the size of the vulnerable population, thereby limiting the spread of a worm outbreak. In the Internet context, the vulnerability of the population is a function of the software engineering practices that produce security vulnerabilities as well as the socio-economic conditions that ensure the homogeneity of the software base. For example, a single vulnerability in a popular software system can translate into millions of vulnerable hosts. While there is an important research agenda, initiated in [3]–[6], to increase the security and heterogeneity of software systems on the Internet, we believe that widespread software vulnerabilities will persist for the foreseeable future. Therefore, pro-active prevention measures alone are unlikely to be sufficient to counter the worm threat.

B. Treatment

Treatment technologies, as exemplified by the disinfection tools found in commercial virus detectors [7] and the system update features in popular operating systems [8], are an important part of any long-term strategy against Internet pathogens. By deploying such measures on hosts in response to a worm outbreak, it is possible to reduce the vulnerable population (by eliminating the vulnerability exploited by the worm) and reduce the rate of infection (by removing the worm itself from infected hosts). However, for practical reasons, these solutions are unlikely to provide short-term relief during an acute outbreak. The time required to design, develop and test a security update is limited by human time scales – usually measured in days – far too slow to have significant impact on an actively spreading Internet worm. Worse, if the installation of such updates is not automated, the response time can be substantially longer. For example, during the Code-Red epidemic it took sixteen days for most hosts to eliminate the underlying vulnerability and thousands had not patched their systems six weeks later [1]. Finally, creating a central authority for developing, distributing, and automatically installing security updates across hundreds of thousands of organizations will require a level of trust and coordination that does not currently exist [9].

C. Containment

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

There are strong reasons to believe that containment is the most viable of these strategies. First, there is hope that containment can be completely automated, since detecting and characterizing a worm – required before any filtering or blocking can be deployed – is far easier than understanding the worm itself or the vulnerability being exploited, let alone creating software to patch the problem. Second, since containment can potentially be deployed in the network it is possible to implement a solution without requiring universal deployment on every Internet host.

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

From our simulation experiments, we conclude that it will be very challenging to build containment systems that prevent widespread infection from worm epidemics. In particular, we find that for such systems to be successful against realistic worms they must react automatically in a matter of minutes and must interdict nearly all Internet paths. Moreover, future worms increase these requirements dramatically, and for most realistic deployment scenarios there are aggressive worms that cannot be effectively contained [9].

The remainder of this paper is organized as follows. Section II discusses the background of worm epidemics, and Section III develops our basic model and methodology for simulating worm growth and worm containment systems. Section IV evaluates this model in an idealized, universal deployment scenario, while Section V extends this to realistic deployment scenarios. Finally, we conclude in Section VI.

II. Background

The term "worm" was first coined in 1982 by Shoch and Hupp of Xerox PARC [10]. Inspired by the "tapeworm" program described in John Brunner's 1972 novel, "The Shockwave Rider", Schoch and Hupp used the term to describe a collection of benign programs that propagated through a local area network performing system maintenance functions on each workstation they encountered. The security implications of self-replicating code were not explored by researchers until 1984, when Fred Cohen described the initial academic experiments with computer viruses in his 1984 paper "Computer Viruses – Theory and Experiments" [11]. However, the Internet worm of 1988 was the first well-known replicating program that self-propagated across a network by exploiting security vulnerabilities in host software. This program, which infected several thousand hosts and disrupted Internet-wide communication due to its high growth rate, is the modern archetype for contemporary Internet worms [12], [13].

There have been few studies of computer worms since 1988, perhaps because there have been few outbreaks until recently. However, in response to Code-Red several quantitative studies of its growth have been developed. Staniford-Chen et al. provide an analytic model of Code-Red's growth matched to empirical observations [9]. Moore and Shannon have also published an empirical analysis of Code-Red's growth, repair, and geography based on observed probes [1] to a dedicated class A network (similar to that described in [14]). Song et al. reproduced parts of this study and further distinguished between different worms simultaneously active [15].

Code-Red has also inspired several countermeasure technologies. One such project, La Brea, attempts to slow the growth of TCP-based worms by intercepting probes to unallocated addresses and artificially placing such connections in a persistent state [16]. In such a state, the thread that was used to initiate the probe will be blocked (Code-Red and other worms are typically multi-threaded) and therefore the worm's rate of infection will decrease. However, it is unclear how effective this approach is even under idealized circumstances, and it is unfortunately easily circumvented by modifying a worm to operate asynchronously. A more compelling approach for slowing the spread of a worm is the per-host "throttling" described by Williamson [17]. Under this scheme, each host restricts the rate at which connections to "new" hosts may be issued. If *universally* deployed, this approach can reduce the spreading rate of a worm by up to an order of magnitude, while not unduly impacting most legitimate communications – however the overall exponential growth pattern of the worm will remain unchanged. To contain the spread of a worm, Toth et al. propose a system for automatically detecting infected hosts within an enterprise network and using firewall filters to prevent them from spreading further (by blocking access to affected ports) [18]. While this strategy by itself is ineffective at containing an epidemic, the constituent technologies could be used in other general containment solutions. Finally, a network technology that was utilized to help block the spread of Code-Red was Cisco's Network Based Application Recognition (NBAR) feature [19]. NBAR allows a router to block particular TCP sessions based on the presence of individual strings in the TCP stream. By filtering on the stream's contents rather than just the header, NBAR allowed sites to block inbound worm probes while still providing public access to their Web servers. Similar functionality is increasingly available in modern switch and router designs and could form the basis of a future containment system.

Several researchers have also examined alternative worm spreading approaches. While Code-Red used a uniform random probe strategy, Code-Red-II was designed to prefer hosts in the same address prefix. A far more virulent approach was proposed by Nicholas Weaver, who described "Warhol Worms" that explicitly choose a set of foundation hosts to infect (based on earlier reconnaissance) and partition this set among replicas to infect the population more quickly [20]. Expanding on this study, Staniford et al. describe "Flash Worms" that contain a complete list of hosts to infect [9]. Rough analytic estimates for these worms suggest that they could exceed the degree of infection of Code-Red in a matter of a few minutes or less (compared to over a dozen hours for Code-Red). Finally, Stanford et al. also describe "Surreptitious Worms" that hide in existing communications patterns to avoid detection. While the behavior of these worms is consistent with our analysis, the orders of magnitude increase in incidence of Warhol and Flash worms and the stealthy nature of Surreptitious worms may invalidate most practical approaches for detecting and responding to a new outbreak.

The work that is perhaps the closest to our own comes from the epidemiological analysis of computer viruses. Kephart and White provide perhaps the most complete analysis of computer virus spread based on random graph topologies. They show that limited defenses are effective as long as the infection rate does not exceed a critical threshold [21]. More recently, Wang et al. have analyzed the impact of immunization on the spread of computer viruses [22] using a similar model. Our work is distinct from these in several dimensions. First, we use real empirical data about host susceptibility, network topology and administrative structure to describe how worms spread on the real Internet. Second, worms are qualitatively different from viruses because they don't require human intermediation to spread. As a consequence, worms typically produce an infection rate many orders of magnitude faster than traditional viruses, while any treatment mechanisms are applied at roughly the same rate. This observation invalidates most of the threshold assumptions in most previous work oriented towards computer viruses. Finally, this same high-speed growth leads us to focus on containment-based approaches that are not explored in the traditional computer virus literature.

N	size of the total vulnerable population
$S(t)$	susceptibles at time t
$I(t)$	infectives at time t
β	contact rate
$s(t)$	susceptibles ($S(t)$) / population (N) at time t
$i(t)$	infectives ($I(t)$) / population(N) at time t

TABLE I
COMPONENTS OF THE SI MODEL.

III. BASIC MODEL

A. *Modeling Worms*

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

This model, described in considerably more detail in [23], dictates that the number of new infections (or incidence) caused by a pathogen is determined by the product of the number infected individuals (infectives), the fraction of uninfected individuals (susceptibles) and an average contact rate. More formally, using the terms defined in Table I, we say the SI model is defined by:

$$\frac{dI}{dt} = \beta \frac{IS}{N}$$

$$\frac{dS}{dt} = -\beta \frac{IS}{N}$$

which can be rewritten as follows:

$$\frac{di}{dt} = \beta i(1-i)$$

Solving this equation, for some constant of integration T, describes the proportion of infected individuals at time t:

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

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

This result is well known in the public health community and has been thoroughly applied to digital pathogens as far back as 1991 [21]. To apply this result to Internet worms, the variables simply take on specific meanings. The population, N, describes the pool of Internet hosts vulnerable to the exploit used by the worm. The susceptibles, $S(t)$, are hosts that are vulnerable but not yet exploited, and the infectives, $I(t)$, are computers actively spreading the worm. Finally, the contact rate, β, can be expressed as a function of worm's probe rate r and the targeting algorithm used to select new host addresses for infection.

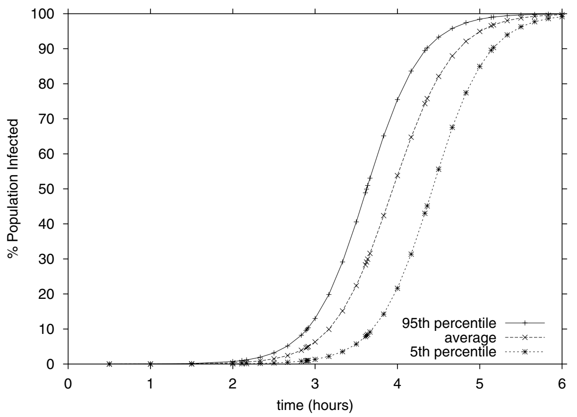


Fig. 1. The simulated propagation of Code-Red-like worms showing the relationship between the average fraction of the vulnerable population infected and the 5th and 95th percentiles.

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

There are two important caveats to this model, both arising from the use of a single scalar variable β to represent the transmission of the worm between infective and susceptible. First, it does not directly account for preferential targeting algorithms. Several recent worms, such as Code-Red II [1], [24], [25] and Nimda [26], [27], preferentially select targets from address ranges closer to the infected host (in the same /24 or /16 network). Similarly, it is difficult to constructively estimate β for the intentional targeting algorithms described by Staniford et al [9]. However, in both cases these worms produce results that can be simply modeled by a direct scaling of β.

A second limitation is that β expresses the *average* contact rate per unit time and does not capture the impact of early variability in the targets selected. Consequently, while the results estimate the growth of the average worm, a particular epidemic may grow significantly more quickly by making a few lucky targeting decisions early on. A worm propagates by probing hosts at rate r and, on each probe, targets a susceptible host with probability $N/2^{32}$ on average. However, for a given worm outbreak, the worm might probe susceptible hosts with higher probability merely by chance as it starts to spread. If the worm manages to infect more susceptible hosts early on than average, then it will spread at a higher rate than the average rate. As a result, the worm will infect the susceptible population more quickly than average. There even exists the possibility, albeit with very low probability, that a worm could always target a susceptible host on each probe as it spreads, in which case the worm would spread at a maximum contact rate of $\beta = r$.

The effects of variability in worm propagation can be significant and a straightforward average-case analysis can obscured them. For example, Figure 1 plots the results of 100 simulations of the propagation of a Code-Red-like worm. The graph shows the percentage of susceptible hosts that a worm infects as a function of the time the worm is allowed to propagate. We plot three different summaries of the 100 simulations: the average case, and the 5th and 95th percentiles. From the graph we see that, after four hours of propagation, the worm infects 55% of susceptible hosts on average. In contrast if we desire 95% confidence then we can only say that, in 95 out of 100 worm outbreaks, up to 80% of susceptible hosts are infected, significantly more than the average case.

While no containment system can prevent all possible permutations of a worm's propagation, we believe that designing for the average case is inadvisable since such a system will fail with regularity. For this reason, the remainder of this paper relies exclusively on simulation results that use the 95th percentile of population infected as determined from a minimum of 100 simulations.

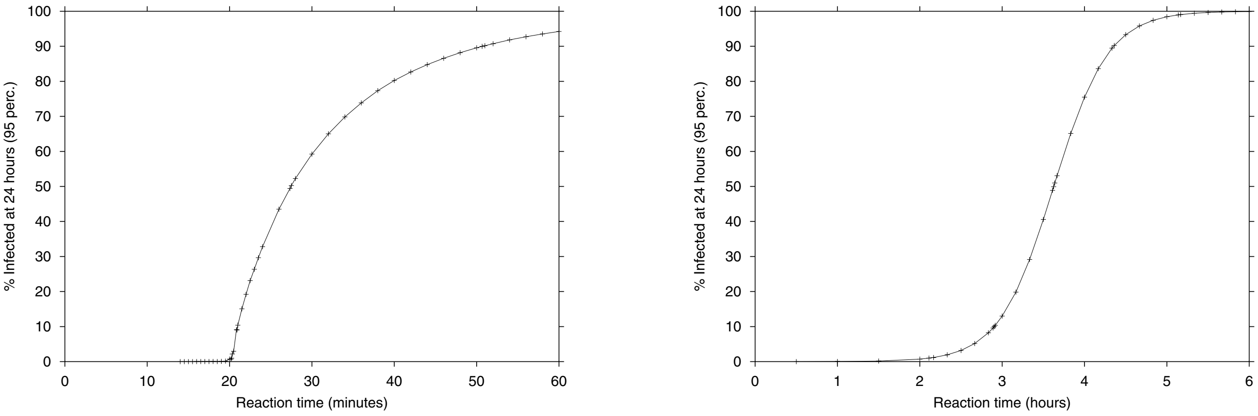
B. Modeling Containment Systems

To understand how various containment techniques influence the spread of self-propagating code, we simulate three factors that determine the ultimate prevalence of the worm: reaction time, containment strategy, and deployment scenario.

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

Containment strategy. The containment strategy refers to the particular technology used to isolate the worm from susceptible hosts. We focus on two key strategies: *address blacklisting* and *content filtering*. The former approach, similar to that used by some anti-spam systems, requires a list of IP addresses that have been identified as being infected. Packets arriving from one of these addresses are dropped when received by a member of the containment system. This strategy has the advantage that it can be implemented with today's filtering technology, does not require the worm to be identified and has a predictable effect on traffic from a given host. However, it must be updated continuously to reflect newly infected hosts and if the detection technology produces false positives then this approach can unintentionally deny service to uninfected nodes.

The second approach requires a database of content signatures known to represent particular worms. Packets containing one of these signatures are similarly dropped when a containment system member receives one. This approach requires additional technology to characterize worm outbreaks and automatically create appropriate content signatures. However, it has the key advantage that a single update is sufficient to describe any number of instances of a particular worm implementation. This approach also includes the possibility for unintended denial-of-service, although it is unlikely for



(a) Address Blacklisting

(b) Content Filtering

Fig. 2. Propagation of the Code-Red worm as a function of reaction time using the (a) address blacklisting and (b) content filtering strategies.

well-chosen signatures, and depends on the assumption that the worm itself is not polymorphic[1].

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

Finally, while some combinations of parameters are sufficient to stop the spread of a worm indefinitely, others simply slow its growth. To capture the impact of this latter effect, we must limit our analysis to some finite time period. In this paper, we evaluate the success of each containment system design based on the outcome occurring after 24 hours. While this value is somewhat arbitrary, we believe it represents a fair lower bound on the time for highly motivated specialists to develop and deploy treatment protocols for eliminating the worm from infected systems. Clearly, experimental evidence collected during the Code-Red epidemic indicates that human system administrators are not able to routinely intervene in any *less* than a 24 hour period [1].

IV. Idealized deployment

In this section we explore the interaction of worm incidence and containment parameters in an idealized baseline setting in which the containment system is universally deployed and information about worm infections is distributed everywhere simultaneously. In this "best case" scenario, every non-infected host implements the chosen containment strategy immediately upon being notified of an infection. This simplified setting allows us to understand the true lower bounds on containment and better understand the fundamental tradeoffs. However, we revisit and remove the universal deployment assumption in Section V.

A. Simulation Parameters

For this baseline analysis, we chose worm parameters based on the Code-Red v2 spread described in [1]: the simulator manages 360,000 total vulnerable hosts out of a total population 2^{32} and the probe rate defaults to 10 per second. We assume that any probe from an infected host to a susceptible host produces an infection immediately. A probe to a non-vulnerable host or a host that is already infected has no effect.

In simulating the containment system we model reaction time as follows: The first "seed" hosts are infected at time 0 and begin to probe randomly. If a host is infected at time t we assume that all susceptible hosts are notified of this fact at time $t + R$, where R specifies the reaction time of the system. When using address blacklisting, this notification simply consists of the IP address of the infected host. Probes from the infected hosts will be ignored from that time forward. Similarly, in content filtering systems this notification simply includes the signature of the worm, and all worm probes from *any* host are ignored afterward. Our goals are to determine the reaction times necessary to minimize worm propagation, to compare the effectiveness between containment strategies, and to understand the relationship between reaction time and the aggressiveness of worm propagation.

B. Code-Red Case Study

As a first step, we examine the effectiveness of this idealized containment system on a Code-Red-style worm. While future worms are likely to be more severe, we argue that any containment system must at least mitigate a worm of this magnitude. We start with two basic questions: How short a

[1] A polymorphic worm is one whose payload is transformed regularly, so no single signature identifies it. In the limit, such a worm could require a unique signature per infected host and content filtering would behave equivalently to address blacklisting.

reaction time is necessary to effectively contain the worm? And, how do the two containment strategies, blacklisting and filtering, compare in terms of behavior and effectiveness?

Figure 2 shows the results of simulations using (a) address blacklisting and (b) content filtering. In these figures, we show the effectiveness of the containment strategy as a function of the reaction time. We measure effectiveness in terms of the percentage of susceptible hosts that become infected after 24 hours with 95% certainty. We consider an approach to be completely effective if it limits infection to 1% of the hosts within the 24 hour period.

In Figure 2(a), we see that address blacklisting is completely effective with any reaction time less than 20 minutes. Within this time frame, the containment system reacts faster than the expected time for an infected host to find and infect a susceptible host on the Internet. As a result, the containment system detects and blacklists all infected hosts before they have a chance to propagate. With larger reaction times, however, the system crosses a threshold where the expected time to locate a new susceptible host is smaller than the reaction time, allowing the worm to continue spreading. Initially, the growth in infections is dramatic (at 20–30 minutes) since most susceptible hosts are not yet infected and the probability of an infected host probing a susceptible host is maximized. As reaction time increases, though, the rate of growth in worm propagation decreases because the number of susceptible hosts decreases as the worm propagates. Even though the containment system takes longer to react to the worm, it takes increasingly longer for infected hosts to find and infect the remaining susceptible hosts with random probes. Eventually, though, with a large enough reaction time the worm will infect all vulnerable hosts within the 24 hour period; although not shown, this happens with a reaction time of 2 hours or longer. As well, over an infinite time window (greater than 24 hours) any reaction time longer than 20 minutes will allow the worm to eventually infect all vulnerable hosts.

In contrast, in Figure 2(b) we see that the content filtering strategy is very different in both effectiveness and behavior. In terms of effectiveness, content filtering prevents the worm from spreading with a reaction time of less than 2 hours, a factor of six difference compared to blacklisting. Moreover, the shape of the content filtering curve is very different. In this scenario, the worm propagates unchecked until the reaction time elapses. Once the reaction time has elapsed, however, content filtering immediately halts further propagation of the worm. As a result, Figure 2(b) shows the typical S-shape of idealized infection curves and is equivalent to the 95% percentile curve shown earlier in Figure 1.

In summary, while both address blacklisting and content filtering can be effective in containing a Code-Red-style worm (assuming idealized deployment), content filtering is significantly more efficient because the first infection provides sufficient information to block all future infection attempts.

C. *Generalized Worm Containment*

The previous section indicates that Code-Red-style worms can be contained assuming universal deployment using systems with plausible reaction times. However, while a significant threat, Code-Red represents only one class of worms. Newer worms like Nimda [26], and hypothesized worms like "Flash Worms" [9], have the potential to propagate much more aggressively than Code-Red. Given the likelihood of more aggressive worms in the future, in this section we estimate the engineering requirements for containing these as well.

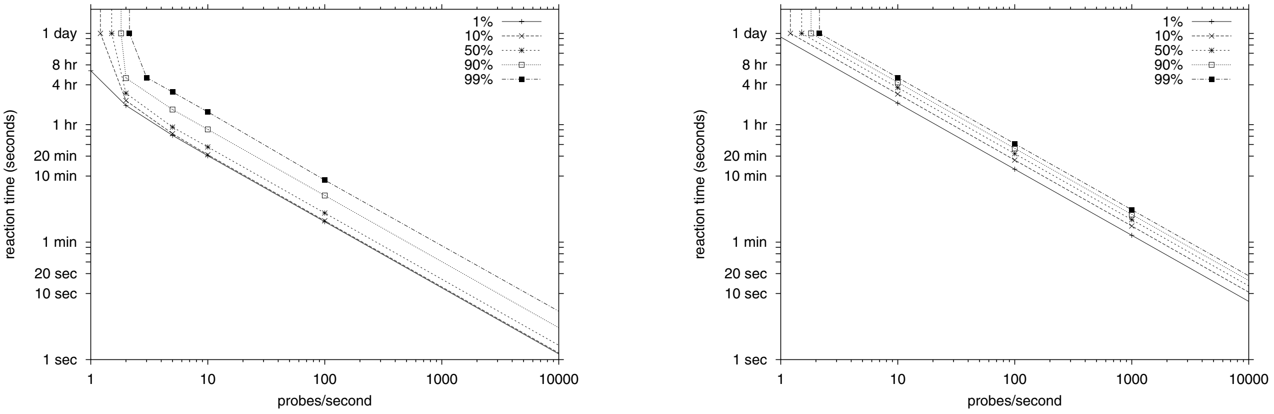
We answer this question by analyzing the relationship between containment effectiveness and worm aggressiveness. We represent effectiveness as the reaction time required to contain a worm to a given degree of global infection, and we represent worm aggressiveness using the rate at which an infected host probes others to propagate the worm. This representation uses a single metric to naturally capture various mechanisms by which worms can increase their infection rate, such as improving their probing implementations, biasing nearby targets within their network, or optimizing the algorithm by which the range of susceptible hosts are partitioned and targeted.

Figure 3 shows the results for the two containment strategies, (a) address blacklisting and (b) content filtering. In these figures, in log-log scale the x-axis shows the probe rate of the worm, the y-axis shows the reaction time, and each curve corresponds to a particular degree of infection. For example, consider the "10 probes/second" point on the "50%" curve in Figure 3(b). This point indicates that a system using content filtering can limit a 10 probes/sec worm from infecting more than 50% of susceptible hosts so long as it detects and reacts to the worm within 4 hours. Note that the points on the infection curves for 10 probes/second, the probe rate of Code-Red, are effectively samples from the graphs in Figure 2. The spacing between the infection curves corresponds to the growth in worm propagation as a function of reaction time[2]. Just as the difference in reaction time between 1% and 10% infection in Figure 2(a) is small, so is the separation of the 1% and 10% infection curves in Figure 3(a).

We further partition the graphs into three regions. We call the region to the left of the 10% infection curve "Well Contained". In this region, the reaction time is fast enough relative to the probe rate to contain worms to within 10% of the hosts. Similarly, we refer to the region between the 10% infection curve and the 90% infection curve as "Partially Contained", and the region to the right of the 90% infected curve as "Uncontained." The particular choice of these partition points is arbitrary, but they represent one reasonable qualitative evaluation of the data.

These graphs make two important results clear. First, address blacklisting continues to require a significantly larger

[2] Note that the infection curves slope up at very low probe rates (e.g., 1 probe/sec in Figure 3(a)). This behavior is an artifact of the finite time for which we simulate the propagation of the worm (24 hours). At these very low probe rates, after 24 hours the worm is still spreading to susceptible hosts. If we had simulated the worm indefinitely, the curves would remain straight lines at these probe rates.



(a) Address Blacklisting

(b) Content Filtering

Fig. 3. Reaction times necessary for (a) address blacklisting and (b) content filtering to contain worms of various degrees of aggressiveness, as represented by their probe rates. Each curve corresponds to a particular degree of infection (e.g., "10%" refers to 10% of susceptible hosts infected after 24 hours). Note the use of log scales on both axes.

reaction time than content filtering. For worms propagating at 100 probes/second, an order of magnitude faster than Code-Red, an address blacklisting system will have to react in 3 minutes to limit infection to 10%, whereas a content filtering system can take 18 minutes to react while achieving the same level of protection. Given the significant advantage of content filtering over address blacklisting, in the rest of the paper we focus only on content filtering.

The second important observation is that highly aggressive, but realistic, worms require extremely challenging reaction times even for content filtering containment systems. For example, a worm propagating at 1000 probes/second requires a 2 minute reaction time if content filtering is to contain it. Recalling that this simulation still assumes universal deployment, it is easy to see that 1–2 minutes is a minimal threshold for any detection and reaction mechanism. In the next section, we remove the universal deployment assumption and explore the challenges presented by varying deployment scenarios in which content filtering can only intervene on some of the Internet paths between infected and susceptible hosts.

V. Practical deployment

In the previous section, we explored the effectiveness of worm containment under idealized deployment conditions. In practice, however, any containment system is likely to be deployed in a far more limited fashion. In this section, we extend our simulation methodology to include a realistic network model and evaluate the impact of partial deployments on the effectiveness of containment. We are careful to select realistic deployment scenarios in which some fraction of customer networks implement containment at their Internet border, or some number of the highest-connectivity ISPs do the same at their exchange points.

A. Network Model

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

To identify the set of ASes in the Internet and their connectivity, we used the routing table for July 19, 2001 08:00 PDT from the popular Route Views service [28]. This routing table enables us to build an AS topology graph of the Internet using the 11,582 ASes contained in the table. We chose this day and time to reflect the state of Internet routing when the Code-Red worm started propagating.

For a representative set of vulnerable Internet hosts distributed across the Internet, we use the hosts infected by the Code-Red v2 worm during the initial 24 hours of propagation [1]. This set of hosts is large, well distributed throughout the Internet address space, and known to represent hosts with a common vulnerability. We model the topological location of each host by placing it in its origin ASes according to the Route Views data mentioned previously. Note that, because some host IP addresses map to multiple origin ASes, we cannot accurately associate them with a particular origin AS and therefore remove them from consideration. As a result, we include only 338,652 vulnerable hosts distributed among 6,378 ASes when using this network model.

We model different deployment scenarios by assigning groups of ASes to the containment system. It is assumed that an AS belonging to this system can choose to filter any packet passing through one of its routers. To model which packets pass through each AS, we compute the shortest path

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

TABLE II

PATH COVERAGE AMONG VULNERABLE ASES AND END HOSTS.

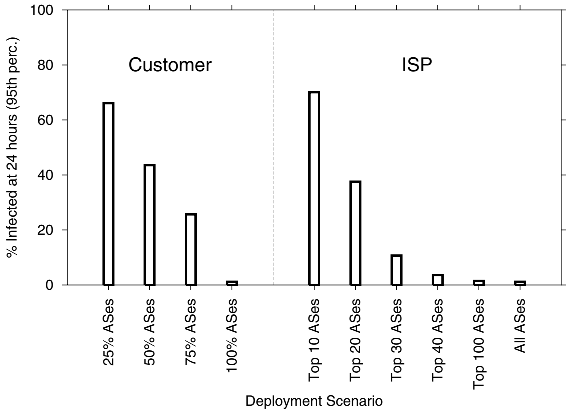


Fig. 4. Containment effectiveness as a function of deployment scenario.

on the graph of all AS adjacencies in the routing table. In the absence of policy, BGP will choose the shortest AS path of all equal cost options. However, we found that many pairs of ASes were connected by multiple equal-cost shortest paths (with an average of 6.3 equal-cost paths for every AS pair). We explored several different techniques to resolve such ties and observed no significant differences between them. For the remainder of this paper, we break ties by selecting the AS path with the greatest outdegree sum.

B. *Deployment Scenarios*

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

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

C. *Code-Red Case Study*

Figure 4 shows the effectiveness of containment for various configurations of the two deployment scenarios using the original Code-Red parameters and the content filtering containment strategy. We select a reaction time of 2 hours, which contains the worm to less than 1% of vulnerable hosts in the idealized deployment scenario described earlier. The y-axis of the graph shows the fraction of all vulnerable hosts that become infected 24 hours after the start of worm propagation.

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

The bars on the right of Figure 4 show various degrees of deployment using the Top ISP deployment scenario. In this scenario, the N largest ISPs block all worm probes traversing their networks, including probes on incoming, outgoing, and transit flows. From these results, we see that a worm can be contained to a minority of hosts if the top 20 ISPs cooperate, and including the top 40 ISPs is sufficient to limit the worm to less than 5% of all hosts.

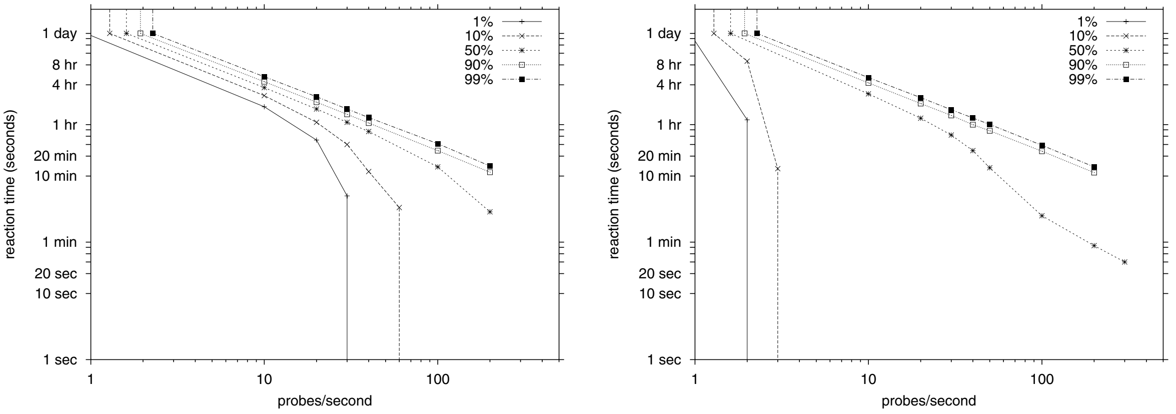
The advantages of the ISP approach relates directly to their role in carrying transit traffic. As seen in Table II, filtering traffic at the customer networks contained within 75% of all ASes intercepts less than 75% of all potential paths. By contrast, the top 10 ISPs alone can interdict almost 90%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

D. *Generalized Worm Containment*

We have seen the effects of more realistic deployment scenarios on the propagation of a Code-Red worm. In this section, we study the reaction time requirements of these deployment scenarios on more aggressive worms.

As in Section IV-C, we explore the relationship between containment system reaction time and worm aggressiveness. Figure 5 shows the results for two deployment scenarios using content filtering, (a) the “Top-100 ISP” scenario and (b) the “50% Customer” scenario. These graphs are similar to those in Figure 3, but rather than an idealized network they instead use our network model to capture the effects of deployments that in practice cannot block all paths among vulnerable hosts in the network.

These graphs show two important results. First, we see that it is essential to model network effects: when modeling the



(a) Top 100 ISPs

(b) 50% Customers

Fig. 5. The reaction times required for effective worm containment for various worm intensities. Shown are graphs for the two deployment scenarios of worm containment: (a) "Top 100 ISPs" and (b) "50% Customers."

deployment scenarios, containment encounters inherent limits. In these graphs, the regions where worms can be contained are much smaller with the network model than in the ideal case. In fact, the effectiveness of the containment system is bounded where the curves slope down to meet the x-axis. For these curves of low degrees of infection, the system fails to contain the worm with 95% certainty – even when using reaction times less than one second. With the ideal model in Section IV-C we found that using content filtering could contain an aggressive worm spreading at 100 probes/second to 1% of vulnerable hosts with a reaction time of 18 minutes. In contrast, Figure 5 shows that neither deployment scenario can contain worms to a 1% infection at non-trivial worm probe rates at all. The curves in both graphs for a 1% infection slope down well before 100 probes/second, indicating that the system cannot contain a worm with such an aggressive probe rate. At best for a 1% infection, the system can contain a worm propagating at 30 probes/second for the "Top-100 ISPs" scenario, and only 2 probes/second for the "50% Customers" scenario.

For an aggressive worm propagating at 100 probes/second, what is the minimum infection that a containment system can achieve? Using content filtering in the "Top-100 ISPs" scenario we found that, for a worm spreading at 100 probes/second, the containment system cannot prevent it from spreading to less than 18% of the hosts. Although not shown on the graph, a curve for an 18% infection would be the transition between the set of curves that slope down to the x-axis and the curves that extend out to higher probe rates for arbitrarily small reaction times.

The reason why containment cannot achieve low infection rates for aggressive worms is due to the fact that the deployment scenarios do not cover all of the paths among all of the vulnerable hosts. The "Top-100 ISPs" scenario blocks 99.7% of the paths among hosts. However, at high probe rates the worm is able to infect enough vulnerable hosts before it is detected and blocked that it continues to exploit the 0.3% unblocked paths and spread further. Even with reaction times below one second, these 0.3% unblocked paths are enough of a backdoor that the worm can exploit those paths to infect a significant degree of the network in 24 hours. For example, Figure 5(a) shows that a worm can achieve a 10% infection in the "Top-100 ISPs" scenario if it propagates at a rate of 60 probes/second or faster independent of the reaction time of the containment system.

Second, we see that the "Customer" approach to containment is again not nearly as effective as the "Top ISP" approach for low infection rates. Directly comparing the two graphs in Figure 5, the two deployment scenarios behave similarly for infection rates of 50% and above. However, for lower, more interesting infection rates, the "Customer" approach is significantly less effective. The "50% Customer" scenario cannot limit worms to 1–10% infection rates for probe rates as small as 2–3 probes/second; in other words, it cannot even limit a Code-Red worm to a 10% infection rate. This result shows that the effectiveness of containment is very sensitive to which ASes are involved. In the "50% Customers" scenario over 5,000 ASes are filtering worm probes, but this scenario is still much less effective than the "Top-100 ISPs" scenario that only involves 100 ASes.

VI. Conclusion

In this paper, we investigate the use of widespread containment as a mechanism for mitigating network-borne epidemics. We explore a broad class of containment systems in terms of their abstract properties: reaction time, containment strategy, and deployment scenario. Using a susceptible host population inferred from the Code-Red epidemic, and an empirical Internet topology data set, we use simulation to analyze how such a worm would spread under various defenses, ranging from the existing Internet to an Internet using idealized defense technology.

From our simulation experiments, we make the following conclusions about various aspects of containment systems for worm epidemics:

- *Reaction time*: To prevent widespread infection in the Internet, containment systems will *require* automated methods to detect and react to worm epidemics. If containment systems are unable to activate filtering mechanisms within minutes of the start of an epidemic, such systems will be ineffective in the wide area.
- *Containment strategy*: Content filtering is significantly more effective than address blacklisting and can contain worms an order of magnitude more aggressive. To support this capability, we encourage network equipment vendors to provide flexible high-speed packet classification and filtering services – extending into the application layer.
- *Blocking location*: Nearly all of Internet paths, such as those covered by the 100 largest ASes, need to employ content filtering for a containment system to be effective. As a result, cooperation and coordination among ISPs will need to be extensive.

From these results, we conclude that it will be very challenging to build Internet containment systems that prevent widespread infection from worm epidemics. In particular, designing and implementing systems that automatically detect the start of worm epidemics and then invoke distributed algorithms to activate widespread filtering mechanisms on the order of minutes is a daunting task. And the inevitable emergence of significantly more aggressive worms further complicates the problem.

Acknowledgments

We would like to thank kc claffy (CAIDA), Vern Paxson (LBL and ICIR), Duane Wessels (The Measurement Factory), and the anonymous reviewers for their helpful comments and suggestions. Support for this work was provided in part by DARPA FTN Contract N66001-01-1-8933, DARPA NMS Contract N66001-01-1-8909, National Institute of Standards and Technology Grant 60NANB1D0118, CAIDA members, and Cisco under grant FY2002URB and its University Research Program.

References

[1] D. Moore and C. Shannon, "Code-Red: a Case Study on the Spread and Victims of an Internet Worm," in *Proceedings of the 2002 ACM SICGOMM Internet Measurement Workshop*, Marseille, France, Nov. 2002, pp. 273–284.

[2] Computer Economics, "2001 Economic Impact of Malicious Code Attacks," http://www.computereconomics.com/cei/press/pr92101.html.

[3] C. Cowan, C. Pu, D. Maier, J. Walpole, P. Bakke, S. Beattie, A. Grier, P. Wagle, Q. Zhang, and H. Hinton, "StackGuard: Automatic Adaptive Detection and Prevention of Buffer-Overflow Attacks," in *Proceedings of the 7th USENIX Security Conference*, San Antonio, Texas, Jan. 1998, pp. 63–78.

[4] D. Wagner, J. S. Foster, E. A. Brewer, and A. Aiken, "A First Step towards Automated Detection of Buffer Overrun Vulnerabilities," in *Proceedings of the Network and Distributed System Security Symposium*, San Diego, CA, Feb. 2000, pp. 3–17.

[5] G. C. Necula, "Proof-Carrying Code," in *Proceedings of the 24th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL '97)*, Paris, France, Jan. 1997, pp. 106–119.

[6] A. Somayaji, S. Hofmeyr, and S. Forrest, "Principles of a Computer Immune System," in *New Security Paradigms Workshop*, Sept. 1997, pp. 75–82.

[7] Symantec, "Symantec Security Response," http://securityresponse.symantec.com/.

[8] Microsoft Corporation, "Microsoft windows update," http://windowsupdate.microsoft.com.

[9] S. Staniford, V. Paxson, and N. Weaver, "How to 0wn the Internet in Your Spare Time," in *Proceedings of the 11th USENIX Security Symposium*, San Francisco, CA, Aug. 2002.

[10] J. F. Shoch and J. A. Hupp, "The Worm Programs: Early Experience with a Distributed Computation," *Communications of the ACM*, vol. 25, no. 3, pp. 172–180, March 1982.

[11] F. Cohen, "Computer viruses — theory and experiments," *Computers and Security*, vol. 6, pp. 22–35, 1987.

[12] J. Rochlis and M. Eichin, "With Microscope and Tweezers: The Worm from MIT's Perspective," *Communications of the ACM*, vol. 32, no. 6, pp. 689–698, June 1989.

[13] E. Spafford, "The Internet Worm: Crisis and Aftermath," *Communications of the ACM*, vol. 32, no. 6, pp. 678–687, June 1989.

[14] D. Moore, G. M. Voelker, and S. Savage, "Inferring Internet Denial-of-Service Activity," in *Proceedings of the 10th USENIX Security Symposium*, Aug. 2001, pp. 9–22.

[15] D. Song, R. Malan, and R. Stone, "A Snapshot of Global Internet Worm Activity," Arbor Networks, Tech. Rep., Nov. 2001.

[16] T. Liston, "Welcome To My Tarpit: The Tactical and Strategic Use of LaBrea," Tech. Rep., 2001, http://www.threenorth.com/LaBrea/LaBrea.txt.

[17] M. Williamson, "Throttling Viruses: Restricting Propagation to Defeat Malicious Mobile Code," HP Laboratories Bristol, Tech. Rep. HPL-2002-172, 2002.

[18] T. Toth and C. Kruegel, "Connection-history Based Anomaly Detection," in *Proceedings of the IEEE Workshop on Information Assurance and Security*, West Point, NY, June 2002.

[19] Cisco Systems, Inc, "Using Network-Based Application Recognition and Access Control Lists for Blocking the "Code Red" Worm at Network Ingress Points," http://www.cisco.com/warp/public/63/nbar˙acl˙codered.shtml.

[20] N. C. Weaver, "Warhol Worms: The Potential for Very Fast Internet Plagues," http://www.cs.berkeley.edu/˜nweaver/warhol.html.

[21] J. O. Kephart and S. R. White, "Directed-Graph Epidemiological Models of Computer Viruses," in *IEEE Symposium on Security and Privacy*, 1991, pp. 343–361.

[22] C. Wang, J. Knight, and M. Elder, "On Computer Viral Infection and the Effect of Immunization," in *Proceedings of the 16th Annual Computer Security Applications Conference (ACSAC'00)*, New Orleans, LA, Dec 2000.

[23] H. W. Hethcote, "The Mathematics of Infectious Diseases," *SIAM Review*, vol. 42, no. 4, pp. 599–653, 2000.

[24] eEye Digital Security, "CodeRedII Worm Analysis," Aug. 2001, http://www.eeye.com/html/Research/Advisories/AL20010804.html.

[25] SecurityFocus, "SecurityFocus Code Red II Information Headquarters," http://aris.securityfocus.com/alerts/codered2/.

[26] Symantec, "W32.nimda.a@mm," http://www.symantec.com/avcenter/venc/data/w32.nimda.a@mm.html.

[27] CERT, "CERT Advisory CA-2001-26 Nimda Worm," http://www.cert.org/advisories/CA-2001-26.html.

[28] D. Meyer, "University of Oregon Route Views Project," http://www.routeviews.org/.