A Hybrid Quarantine Defense

Phillip Porras, Linda Briesemeister, Keith Skinner
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025
{phillip.porras, linda.briesemeister, keith.skinner}@sri.com

Karl Levitt, Jeff Rowe, Yu-Cheng Allen Ting
Department of Computer Science
University of California, Davis
One Shields Avenue
Davis, CA 95616
{levitt, rowe, yting}@cs.ucdavis.edu

ABSTRACT

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.

Categories and Subject Descriptors

C.2 [Computer and Communication Networks]: Security and Protection – Worms; C.2.3 [Network Operations]: Network monitoring – Worm Detection; C.2.5 [Local and Wide-Area Networks]: Internet; C.4 [Performance of Systems]: Modeling Techniques Simulation;

General Terms

Algorithms, Experimentation, Security, Performance

Keywords

Network Security, Network Modeling and Simulation, Worms, Worm Detection Systems

1. INTRODUCTION

In recent years we have witnessed the disturbingly high frequency with which outbreaks of self-propagating malicious code have plagued public networks, and have observed these epidemics penetrate into even well-protected enterprises, particularly as computing assets become more mobile. To combat this problem, there has been a surge of research in developing techniques to recognize and defend networks from emerging malicious code epidemics.

WORM'04, *OCTOBER 29, 2004, WASHINGTON, DC, USA.*

We report on an ongoing study, in which we assess the comparative strengths of complementary quarantine philosophies, and explore the potential benefits of merging them to offer protection that is significantly more effective than either approach alone. Our current study examines two complementary worm quarantine strategies: one relying on autonomous gateway protection devices, and the other relying on peer-based coordinated sharing. Several variations of the algorithms discussed here have been published elsewhere; however, here we focus on comparing the effectiveness of these quarantine strategies across a range of worm infection algorithms.

We also propose a novel hybrid defense, which combines the two complementary quarantine strategies. Our assessment reveals that this hybrid approach offers substantial infection growth rate reductions, greater than either technique can achieve alone. Our results suggest the potential value in developing hybrid quarantine solutions that operate both autonomously at network domains, but can also coordinate to provide group-wide protection.

2. Ongoing Research in Malicious Code Defense

Over the last decade large-scale malicious code epidemics have evolved from rare nuisance applications and research curiosities into the most well recognized information-based global security threat known today. The field of worm countermeasure development is active, with several new and derivative strategies being proposed yearly. Moore et.al. [8] propose various requirements for consideration in developing containment strategies (e.g., network filtering), discussing issues such as reaction time, infection countermeasures, and deployment strategies, and explores how these factors impact worm propagation dynamics.

Substantial effort has been performed in techniques that we classify as *Resource Limiting (RL) Solutions*. RL solutions explore ways in which local systems and domains may delay worm propagation through the limiting of resources that aggressive worms are known to consume at high rates. Williamson [16] suggests that throttling the volume of outbound connections that a host is allowed to initiate to new machines can produce a significant reduction in the infection rate, without significantly hindering normal communications. Staniford [12] refines the outbound connection-throttling concept and provides extensive assessment of its behavior, while moving the throttling mechanism from the individual host to the domain gateway. Gualtieri and Mosse [6] propose to dynamically calculate outbound connection rate limits on a per process basis, through the observa-

tion of connection rates across the total population of processes or from a pre-selected group of known benevolent processes. Ganger et. al. [4] suggest that the analysis of network connections not facilitated through DNS lookups provide a relevant signature for identifying potential worm traffic, and host-layer filters can be directed to such traffic with greater aggressiveness. Wong et. al. [17] explore the application of connection rate limiting to backbone routers, suggesting that the throttling of IP-to-IP connection at the edge offers propagation reduction equivalent to all hosts implementing rate throttling, while offering significant deployment advantages.

A second major direction has been toward the design of cooperative information sharing, either hierarchically or using peer-based models, to help recognize the emergence of a propagating worm and then take coordinated action before the worm can saturate the network. We broadly classify these schemes as *Leap Ahead (LA) Solutions*, as they seek to spread warning to network segments not yet affected, and thus potentially prevent the worm from reaching its full saturation potential, assuming a finite time-to-patch interval. For example, Nojiri et. al. [10] propose a cooperative alert sharing scheme using a "Friends protocol" under which each node (domain gateway) pre-selects a set of friends with which to share worm indicators, and in turn is also selected by other domains to receive reports. Alternatively, Anaganostaki et. al. [1] propose a variation of this sharing scheme called COVERAGE, in which a node randomly selects a set of remote nodes to poll for worm reports at periodic intervals.

Predesigned-Preventative (PP) Solutions refer to approaches designed to disrupt or thwart the discovery of susceptible nodes within an address space, potentially by dynamically altering the connectivity of networks or end nodes in the presence of a propagating threat. Briesemeister et. al. [3] study percolation theory or epidemic spread in artificial scale-free networks to suggest how networks could be designed to delay the spread of propagating malicious code while still maintaining high reliability of network links. Gorman et.al. [5] also examine the use of scale-free properties within the Internet's autonomous system (AS) map, and similarly suggest that the concentration of worm filtering services on the nodes with the highest connection density would yield the greatest return while disrupting the minimum set of network devices. Staniford's work on CounterMalice [12] and Zou's et. al. [18] study of "firewall network systems," explore the pre-placement of devices within an enterprise that could facilitate its rapid isolation into subnetworks, thwarting a worm's ability to propagate by pre-planning segmentation strategy. Provos [11] suggests the deployment of honeypot devices in a network that engage in slow connection dialogs as a method to dramatically slow an aggressive worm's ability to discover susceptible hosts within an address space. Wang et. al. [14] propose the placement of pre-determined filters within end node network stacks, which can be rapidly activated as a first line of defense when vulnerabilities are initially discovered and before patches are installed.

Another variation of worm defense involves an active strategy of interception and rapid patching, which Nicols [9] refers to as "taking to the battle to the worm." We classify these techniques as *Mobile Combat Solutions*. One approach proposes to eliminate propagating malicious mobile code by distributing a mobile self-replicating code module that searches out for signs of a malicious resident code and vaccinates infected machines through patch or other removal method. For example, Toyozumi and Kara [13] present an analysis of predatory vaccination application called Predator. The paper employs the biologically inspired "Lotka-Volterra" equation to model the interaction of the predator-prey relationship between the malicious code and mobile predator vaccination, with the goal of minimizing the number of predators required to eliminate the virus threat. The paper suggests that a small number of good predators, on the order of a few thousand could contain an aggressive large-scale worm such as Code-Red [7]. Nicols [9] explores four active defense propagation models, from simple scanning systems that race against worms to patch susceptible hosts, to sniper worms that behave similarly to the Predator model.

In this study, we examine the complementary nature of two cyber defense strategies: Connection Rate Limiting representing an RL solution, and the Friends protocol, representing an LA solution. Next, we briefly describe the details of these algorithms as applied in our experiment, and discuss the development of a novel combination strategy which overlays both strategies.

2.1 A Resource Limitation Solution using Outbound Connection Rate Limiting

Our resource limitation strategy focuses on limiting the number of outbound nodes that an internal machine may contact per time interval. This strategy is motivated from the observation that during normal operation the volume of outbound connections to unique machines is relatively small, and that this volume generally increases when a machines is infected by a scan-based worm in proportion to the aggressiveness with which the worm seeks susceptible nodes. The assumption holds well for random-scan worms, but not as well for topology, contagion, flash worm victims, or slow spreading worms that may not necessarily spike the outbound communication patterns of their victims.

Figure 1 illustrates the connection rate-limiting algorithm, as implemented in our experiment. Rate limiting is performed at the gateway of each domain, rather than at the individual internal node. Each internal node is allowed to make $\leq$ N outbound connections per time unit. Outbound connections beyond N per time unit are dropped by the gateway. Packet dropping differs from [16] and [17], both of which discuss packet queuing when the threshold is breached. However, queuing may introduce significant practical question. A host can make any number of internal connections without interference, and thus once a worm enters the domain, it may spread to all internal nodes without interference. A threshold limit of N = 10 addresses per time unit is selected as the default parameter for this algorithm. An analysis of alternative static thresholds is explored in [12].

2.2 A Leap-Ahead Solution using the Friends Quarantine Strategy

For our leap-ahead strategy, we implement a variation of the Friends algorithm described in [10]. Figure 2 illustrates the basic concept from the topological perspective. Essentially,

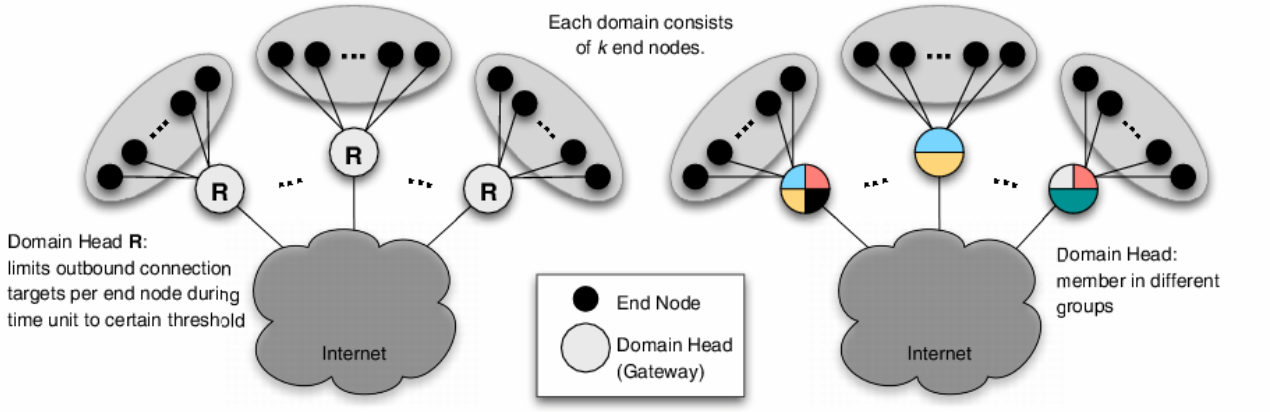

Figure 1 – Connection Rate Limitation and Friends Overview

each domain head (gateway) m ∈ M selects F = G – 1 friends. This selection defines group size G over the population M. The group memberships of one domain head overlap so that one domain head is a member of multiple groups, in which the other domain head selected this one as a friend. This overlap is represented in Figure 2 as the multiple shades of gray at each domain head. Under the Friends protocol, each gateway activates port or content-based filtering, when it receives enough alert from friends (including itself) to indicate the presence of a worm. No single alert is sufficient to trigger filtering, and thus Friends gateways tolerate an adjustable amount of false alarms before they must react to an emerging worm threat. The warning state proceeds to temporally decay until it drops into a state in which filtering is removed from the gateway, but may be raised indefinitely while worm activity indicators persist.

Figure 2 illustrates the details of the Friends algorithm as implemented in our experiment. Briefly, as each worm indicator supplied by the gateway and its friends will impact the gateway's worm suspicion rating by *s*. α represents the alert threshold that must be exceeded before the gateway enters filtering mode, where we calculate α relative to the group size G as follows:

$$\alpha = \left\lceil \frac{G}{4} \quad * \quad s \right\rceil$$

The denominator provides control over the threshold, where higher values increase the aggressiveness with which the gateway enters its defensive posture. Filtering backoff occurs by reducing the suspicion rating by 1 at each time unit. Combined, the α threshold calculation can be adjusted to raise one's guard quickly and to reduce one's defensive posture slowly.

2.3 A Combined Solution using Overlay

Thus far, we have discussed the merits of RL solutions, which seek to slow down aggressive worm propagation by limiting key resources that the worm consumes in order to regenerate itself across the susceptible population. We have also discussed LA strategies, in which network segments share malicious code indicators to gain confidence in the emergence of an outbreak, and then act to prevent the spread of this infection to parts of the network that may not yet be exposed. Intuitively, leap-ahead strategies rely on a collaboration approach which may require several units of time to activate. Thus, in the presence of a leap-ahead defense it is in the worm's strong interest to propagate as rapidly as possible, ahead of the coordinating defenses, and our simulation results demonstrate the effectiveness of this strategy.

We present a combined defense strategy that is inspired by this observation. Here, each gateway will implement a connection-rate limiting defense in parallel with the Friends protocol. The objective is to employ each rate limiter to effectively slow the propagation of aggressive worms, allowing Friends message to propagate to groups and activate a defensive posture in time to halt infection growth before full saturation is reached. The triggering of node rate limiting can itself act as one indicator of worm activity, and extensions of this overlay solutions could include feedback loops in which the rate-limiting threshold may be adjusted by the accumulation of Friends messages at predefined thresholds.

3. Experiment Overview

In this study we employ simulation techniques to examine the macro behavior of connection rate limiting and the Friends peer-to-peer alert sharing algorithms using a symmetric network topology. Several abstractions leave open the questions of infrastructure and communication dynamics that may arise during operational deployments, as we discuss in Section 5. However, before the expense of algorithm implementation and operational testing, simulation provides a time and cost efficient method to bound our expectations of the algorithm before asking directed questions of behavioral properties that arise within specific environments during specific attack conditions.

Our study proceeds with the selection of a base topology and six worm variations comprising fast, medium, and slow-speed, random-scan and topological worms. For this experiment, we have created an automated worm simulation tool suite, which allows us to control and adjust our network topology, epidemic

model assumptions, worm behavior, worm detection quality, and the core parameters of our rate limiter and Friends algorithms. During our simulation phase, we validate our expectations regarding the behavior of our defense algorithms under various worm behavior profiles and under different defense algorithm configurations.

Next we examine the effects of a hybrid combination quarantine strategy, using both our rate limiter and Friends algorithms in parallel. Our study examines the hypothesis that RL and LA strategies are synergistic, such that when combined they can exceed the protection that either strategy provides independently. The remainder of this section presents the network topology employed for our initial assessment, the epidemic models used to test our defense strategies, and the overall experiment hypotheses that we later assess through simulation in Sections 4.

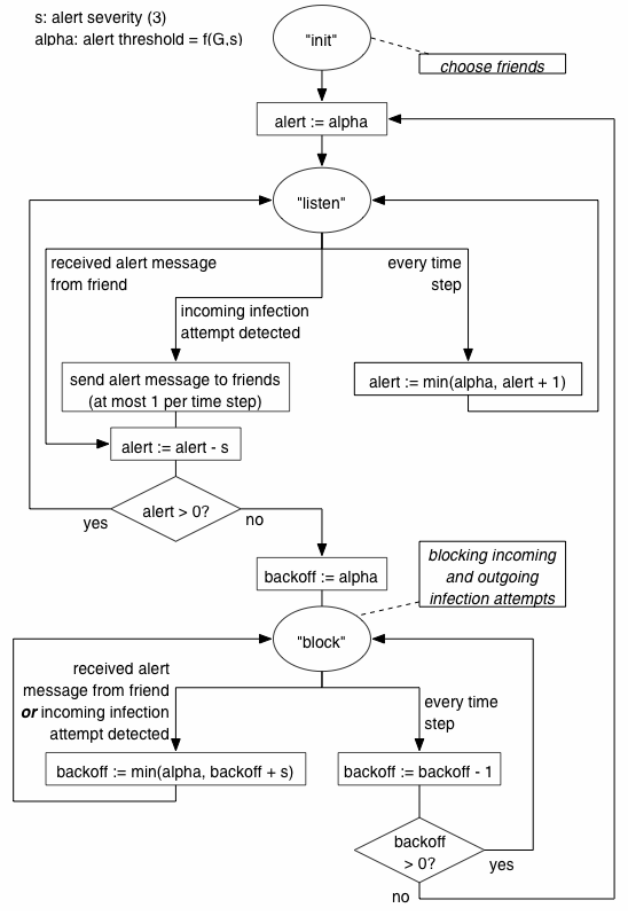

Figure 2 – Flow Diagram of Friends Algorithm

3.1 Topology Overview

Out initial experiments employ a basic network topology consisting of fully connected network domains of equal size, each with a single domain head (or gateway), and a set of end nodes.

We define our base topology through the following parameter:

k: number of end nodes per domain head

M: number of domain heads (gateways)

Thus, the number of end nodes in our network is $k * M$. The total number of nodes (domain heads and end nodes combined) is then $M * (k + 1)$.

In our simulation, all IP addresses are of the form 110.100.X.Y with 110.100.X.1 denoting the gateways. We assume a /16 IP network (address space A $\sim= 2^{16}$) that consists of IP addresses starting with two fixed fields (i.e., given 110.100.X.Y, A is represented in the range X.Y). For domain heads, the value of X is chosen at random from the numbers 2-254. End nodes belonging to a domain head with the IP address 110.100.201.1 for example take the form 110.100.201.Y with Y being chosen at random from the numbers 2-254.

For the experiments reported here we employed the following topological parameters: $k = 10$, $M = 100$. Thus, we have 1000 end nodes and 1,100 total network nodes.

3.2 Epidemic Models

The epidemic model defines the state transitions and conditions under which terminal nodes are transformed by a malicious self-replicating worm. In our model, internal nodes of the network graph represent gateway routers, and for this experiment routing components are not modeled as targets of infection. Our epidemic model distinguishes two states that a terminal node can occupy. First, nodes reside in an initial state of susceptible (S) from attack. When a worm is said to probe an (S) node, the node switches into the infected (I) state.

Our study considers two worm propagation methods. The first method is a random-scan worm, which behaves as follows: per time unit, each infected node n_j selects S random addresses from the address space A and scans the node associated with these addresses. In our experiment, we selected three scan rates: *slow-speed scanner* = 10 scans per unit time, *medium-speed scanner* = 100 scans per unit time, and *high-speed scanner* = 1,000 scans per unit time.

Our second method is a topology-based worm, which differs from random-scan worms in that it prefers to select new targets by interrogating the infected node for information regarding other machines that may share vulnerable services. For example, the worm may use a compromised network service to identify other instantiations of the network services that are scattered throughout the Internet. Alternatively, the worm may seek out infection targets that share a more local network service, such as a vulnerable file- system sharing service. For the purposes of our simulation, we provide a simplified abstraction of the worm by creating an infection target selection process that compromise closely associated machines (e.g., machines within the infect node's domain) with much higher frequency than external machines [15]. We model this behavior using the following algorithm: for each time unit, each infected node n_j selects SL random addresses from the address space A_{local} and SE random addresses from the external address space $A_{external}$. Address space A_{local} represents the local domain (Y = 110.100.X.[2..254], where X represents n_j's domain). Next, we create a scan set $S = \{s_i = [SL \text{ with probability } s\_local \mid SE \text{ with probability } (1\text{-}s\_local)]\}$, and scan the nodes listed in S. As with random-scan worms, our experiment employ four scan rates: 10, 100, and 1,000 scans per time unit, and our default value for s_local = 0.9.

3.3 Experiment Hypothesis

Our selections of cyber defense strategies and worm propagation methods are designed to explore the validity of two conjectures regarding the complementary nature of RL and LA solutions. Figure 3 illustrates three potential worm infection growth trajectories over a plot of time (x-axis) and infection volume (y-axis). S represents the full saturation potential of the network under attack, indicating when the worm has successfully infected all nodes that are susceptible. N represents the moment in time that full saturation is achieved. Trajectory 1 illustrates an example of a reasonably aggressive random infection across an unprotected network.

By limiting the consumption of resources that a worm expends as it searches for susceptible nodes, RL solutions effectively throttle the worm growth rate to a degree that, under ideal circumstances, containment and recovery solutions may be activated to prevent and remove the infection. In this sense, RL solutions impose a delay to saturation, represented by Trajectory 2 reaching S at N′ rather than N. Our expectations regarding the behavior of our connection rate limiter algorithm are shown in Table 1. We expect high-speed random-scan worms will be significantly impacted with respect to saturation delay (N), but perhaps less so to topology worms, which tend to be biased toward its victim's neighbors. Depending on the threshold number of unique outbound connection targets that a node is allowed to initiate per time unit, slow scanning worms may operate at a rate below our connection rate threshold, and thus may not produce an appreciable saturation delay.

	Fast Scan	**Slow Scan**	**Fast Topol**	**Slow Topol**
Rate Limit	Effective (N)	Ineffective	Limited effectiveness (N)	Ineffective
Friends	Ineffective	Effective (S)	Effective (S)	Limited effectiveness (S)
Combo	Effective (N/S)	Effective (S)	Limited effectiveness (N/S)	Limited effectiveness (S)

Table 1 – Summary of Experiment Hypothesis

LA solutions have the potential to prevent a worm from reaching full saturation, at least during the period when all gateways are in their defensive posture. However, consider time *f* in Figure 3. Prior to time *f*, nodes participating in the leap-ahead protocol share worm indicators as each attempts to accumulate enough evidence to initiate a defensive posture. Thus in the presence of a leap ahead strategy, worms do well to propagate aggressively to reach their desired saturation level prior to the group lockdown. Alternatively, a worm that is aware of the backoff parameters of the group could instigate a slow propagation, such that group members never cross the threshold of concern that will allow them to enter their defense posture. However, here all worm instances have to remain silent for multiple time units simultaneously; whereas worm instances must simply slow their rate of connection attempts to avoid the connection rate limiter. Thus, Table 1 suggests that even in slow scanning worms, Friends gateways should be able to recognize and prevent worm spread. As discussed earlier, fast spreading worms can elude the Friends algorithm when they saturate the network faster than the groups can coordinate transitions to defensive postures. Earlier, fast spreading worms can elude the Friends algorithm when they saturate the network faster than the groups can coordinate transitions to defensive postures.

Finally, we bring together the connection rate limiter and Friends algorithms in an experiment to observe their potential synergy. Simply stated, we conjecture that for high-speed worms, the rate limiter has the potential to impose a delay that is significant enough to allow Friends groups to coordinate and block the worm before it can attain its full saturation potential. In this respect, we believe our overlay strategy has the potential to produce a defense that is both efficient with respect to delaying infection growth and preventing full saturation. In addition, we expect the overlay will not harm the effectiveness of either approach, and thus slow scanning worms will at least provide protection equal to the best single defense for this class.

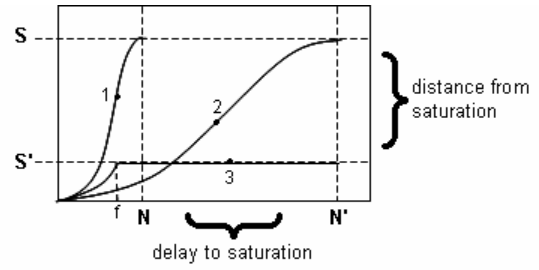

Figure 3 - Countermeasure Effectiveness Metrics

4. Test Simulation Results

Our simulations were conducted using a MATLAB-based simulation environment, which we designed specifically for assessing worm countermeasure performance under diverse network topologies and epidemic models. With respect to topology, our simulator currently supports a parameterized topology generator that can be used to specify the number of domains, and size of each domain, the address space size, and percentage of immune nodes within the network.

Our simulator also provides an extensive degree of flexibility in configuring epidemic models. Currently, our simulator allows the user to select from two possible propagation strategies: random-scan and topology. For our experiment purposes we employed the following epidemic model parameters: All susceptible end nodes are infected when scanned. All end nodes are susceptible. Gateways operate with no false negatives. A run begins with 1 infected node.

The worm simulator currently encodes four worm defense strategies: null defense (to observe the control case), rate limiter, Friends, and the combination or rate limiter with Friends. Figure 4 illustrates our worm simulator's animated graphic display capability and graph generation display. Users can run the worm simulation in text mode, step mode, or play mode, and

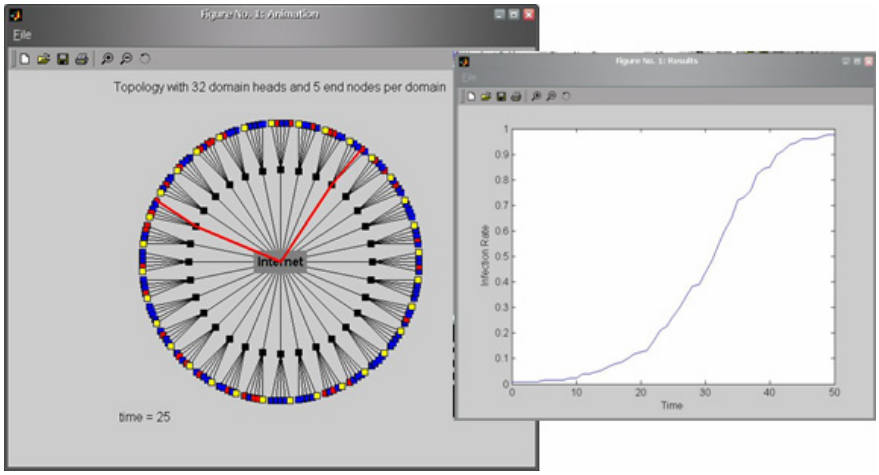

Figure 4 – Worm Simulator Graphic Display

can display of infection attempts and Friends worm indicator messages. The simulator also allows the user to select the time duration that the simulation will span.

This section briefly summarizes our simulation results, which represent 600 simulation runs. We performed 10 runs per combination of worm type and speed and defense algorithm, and averaged the results. Each run lasted for 200 simulated time steps.

	G	50	25	13	7
	(of pop.)	½	¼	~1/8	~1/16
Friends	High-Speed	–	–	–	+
	Medium-Speed	+	+	o	–
	Low-Speed	+	o	–	–
Combo	High-Speed	+	+	o	–
	Medium-Speed	+	o	–	–
	Low-Speed	+	o	–	–

Table 2 – Random-Scan Worms: Group Sizes versus Worm Speeds

4.1 Control Case and Connection Rate Limiter Simulation Results

Figures 5 and 6 illustrate the simulation results for the control case – a defenseless network – and the connection rate limiter defense. All graphs combine the average growth rate of high-speed (1,000 scans per unit time [ptu]), medium-speed (100 scans ptu), and slow-speed (10 scans ptu) worms.

We show infection growth of random-scan and topology-based propagation strategies for the control case and rate limiter defense. The control cases illustrate the average potential growth of worm variations when no defense mechanism is enabled. Subsequent defense simulation results employ the identical network and the above six worm configurations seen in Figures 5 and 6, and these control cases provide the baseline from which to assess the impact of the defense strategy. All rate limiter examples were run with a connection limit threshold that allows each node at most 10 unique outbound connection targets.

In Figure 5, we see that compared to the control case, both high-speed and medium-speed worms slowed the spreading when employing rate limiting. The right most curves are nearly identical; this simulation result is not surprising as the slow-speed worm operates at the rate limit and therefore, and we did not expect to see an influence of the rate limiter defense on this infection growth.

Figure 6 depicts the simulation results of a topology-based propagation strategy. As expected, the rate-limiting defense is nearly ineffective for all worm speeds that spread mainly in the local domain and have only 10% of the infection attempts going through the gateway that employs the resource limitation.

	G	50	25	13	7
	(of pop.)	½	¼	~1/8	~1/16
Friends	High-Speed	–	+	+	–
	Medium-Speed	+	+	o	–
	Low-Speed	+	o	–	–
Combo	High-Speed	+	+	o	–
	Medium-Speed	+	o	–	–
	Low-Speed	+	o	–	–

Table 3 – Topology-Based Worms: Group Sizes versus Worm Speeds

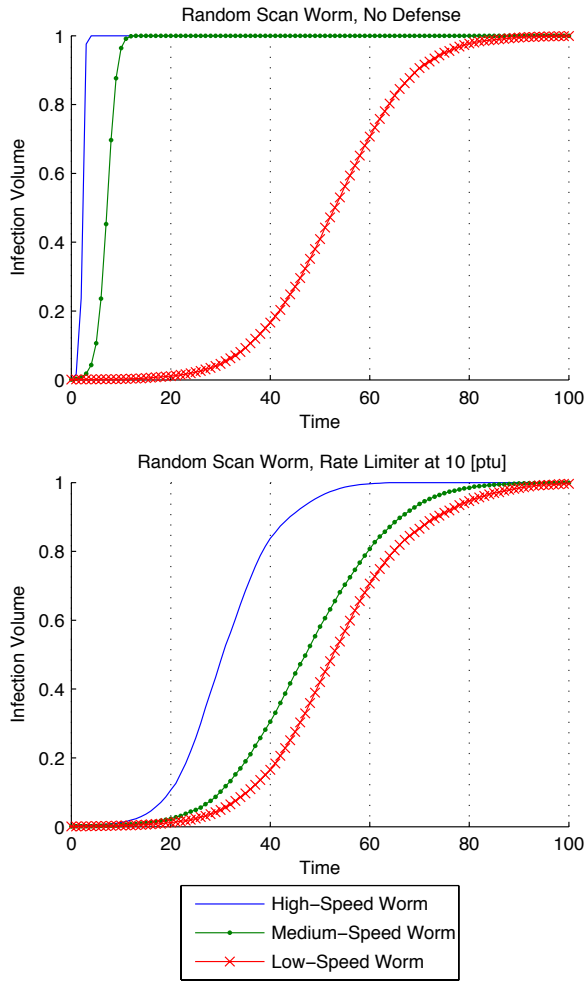

Figure 5 – Results for Random Scan Worm, Defenseless Network and Connection Rate Limiter

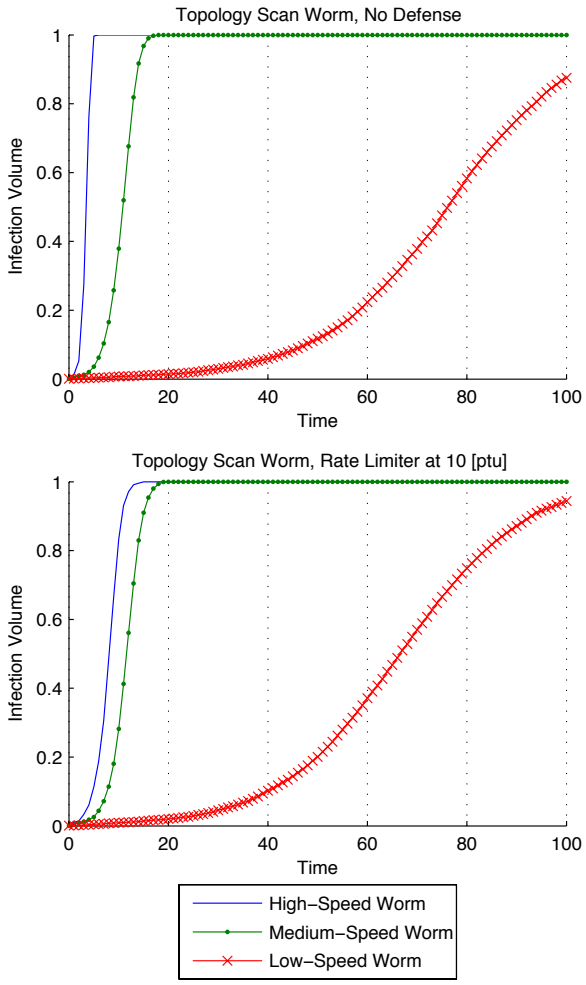

Figure 6 – Results for Random Scan Worm, Defenseless Network and Connection Rate Limiter

4.2 Friends Protocol and Combination Defense Simulation Results

Figures 7 and 8 and Tables 2 and 3 summarize the simulation results for the Friends protocol and combination with rate limiting defenses. Figure 7 and Table 2 address the random-scan worm, and Figure 8 and Table 3 address the topology-based worm propagation strategy. Again, all graphs combine infection growth curves for three different worm speeds. We have varied the group size of Friends throughout the experiments resembling ½, ¼, and approx. 1/8 and 1/16 of the population of gateways that carry out the Friends protocol. Tables 2 and 3 assess the effectiveness of Friends and combination algorithms with respect to these different group sizes and worm speeds for both, random-scan and topology-based worm propagation.

Figure 7 shows simulation results using a group size of G = 25, which is a quarter of the population of gateways. These results show that the network cannot defend against the high-speed random scan worm using the Friends protocol unless it is overlaid with a rate limiting strategy. However, the Friends defense performs almost better against the medium-speed worm than combined with the resource limiting approach. We explain this behavior with the complex dynamics of the Friends scheme. In order to maintain blocking mode, the gateways are stimulated by detecting incoming infection attempts that result from ongoing worm activity in the network, or by receiving alerts that in turn result from ongoing worm activity at other gateways, which hap-pen to have selected the gateway as a friend. Therefore, the Friends defense as simulated in our experiment performs slightly better than combined with rate limiter, affecting outbound

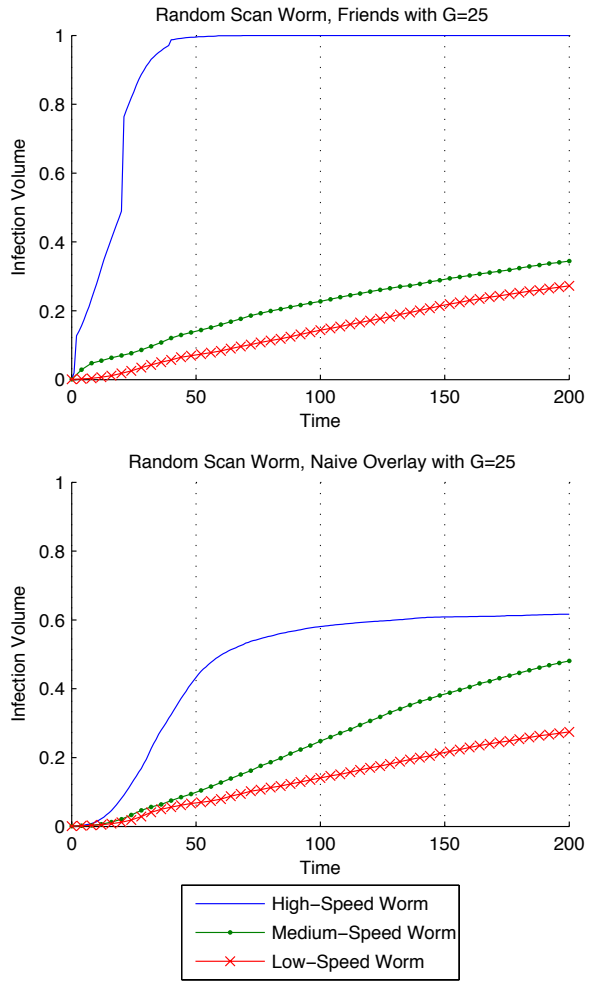

Figure 7 – Results for Random Scan Worm, Friends Protocol and Combination Defense

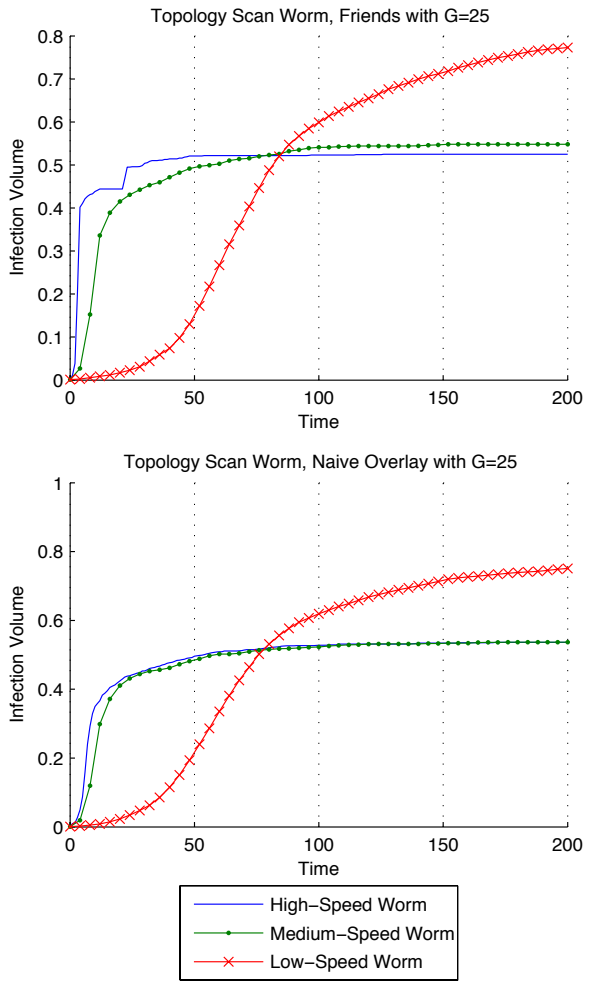

Figure 8 – Results for Topology Scan Worm, Friends Protocol and Combination Defense

infection attempts vital to generating alerts in the system. The curves for the slow-speed worm are again nearly identical as this worm operates below the threshold of the rate limit employed.

Figure 8 illustrates simulation results for defending the network against a topology-based propagation worm using the Friends protocol and a combination approach with rate limiting. Compared to the control case and pure rate limiting, the Friends protocol proves effective in curbing the worm spreading for high-speed and medium-speed worms. The slow-speed worm curves are again nearly identical as the rate limit threshold lies below the scan rate of this worm speed.

Tables 2 and 3 summarize our simulation results qualitatively over different group sizes for various speeds of random-scan and topology-based propagation worms. Our results suggest that tuning the parameters of the Friends algorithm as implemented here affects the defense of a network against different worm propagation strategies. If not combined with a rate limiter, it cannot be said whether a greater group size is favorable or not. For example, a high-speed worm was contained in a number of simulation runs with the smallest group size of G = 4 when using random-scanning. However, for the topology-based propagation strategy and a high-speed worm, neither the smallest nor the largest group size proved to be as effective as group sizes of G = 25 and 13. Our simulation results suggest that the choice of parameters of the Friends strategy is very sensitive to the worm type and speed experience to be effective in its defense purpose.

Only when combined with the rate limiting approach, a larger group size seems favorable over a smaller group size. However, this trend needs to be evaluated under cost measuring performance features such as counting the number of alert messages that generate additional traffic in a network. Other potentially interesting metrics include the number of false positives in a network

simulation with benign background traffic and a less-than perfect detection capability of worm traffic.

4.3 Comparative Results Assessment

The results from the combination defense for high-speed worms show a significant improvement over rate limiting and Friends run independently, as conjectured earlier. The combination approach also makes the choice of the group size parameter proportional to the performance of the defense capability in contrast to employing the Friends protocol alone. Our simulations that varied the group size parameter uncovered very sensitive dynamics of the Friends scheme in correlation to the worm type and speed simulated.

Figure 9 illustrates a merged graph of the infection growth trajectories of all three defense strategies and the control case. Trajectory 1 illustrates the explosive, near immediate, infection potential of the high-speed random-scan worm on the network. Trajectory 2 illustrates Friends' ability to slow and eventually enter a coordinated defense posture, but only after 80% of the susceptible population is infected. Trajectory 3 shows the highly successful impact of aggressive connection rate limiting, which provides a substantial delay in the worm's ability to achieve its full saturation potential. Finally, Trajectory 4 shows our hybrid combination strategy, demonstrating the synergistic effect of using rate limiting to slow the worm, allowing the group coordination power of Friends to prevent this high-speed worm from attaining a significant foothold in the network.

The results from our simulation phase suggest a potentially encouraging research direction in developing hybrid quarantine strategies that leverage the strengths of multiple techniques. The simulation results also provide a basis from which to validate behavioral expectations, and identify optimal algorithm and worm configurations that can subsequently drive testing or emulation experiments.

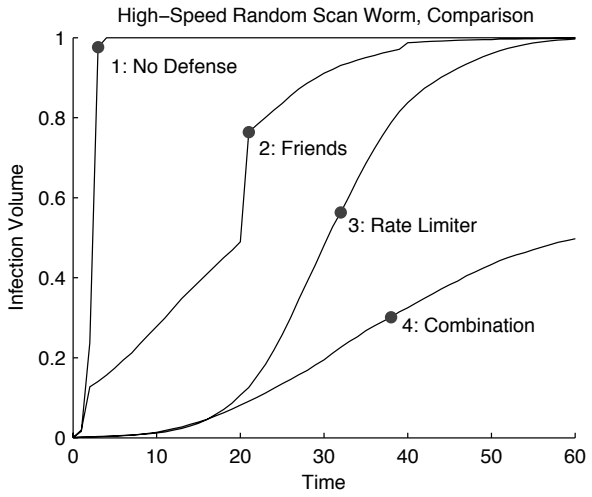

Figure 9 – An Example Infection Growth Rate Comparison Using the High-Speed Random Scan Worm Simulation

5. Discussion and Future Work

A major limitation in the practical deployment of a Friends-based defense is the dependence on accurate alert production services to recognize incoming worm infiltration attempts. Detection inaccuracy have the potential to accelerate or delay the Friend's group defensive posture, and thus the selection of the α threshold plays an important role in deciding the amount of corroboration required before entering the defensive posture. While our worm simulation environment can model false positive and negative rates in alert production, a full exploration of the affects of imperfect detection on corroboration size and infection growth rate is beyond the scope of this paper.

One potential alternative to the alert production questions that arise in Friends is to employ the rate limiter services themselves as the primary source of Friends alert production. That is, a domain's rate limiter and Friends service may interoperate. Each rate limit threshold crossing can be shared among Friends peers as a potential worm threat indicator. Under this scheme, Friends need not rely on inbound intrusion detection services to identify worm traffic. We have considered multiple models of RL/LA integration strategies, including rate limiting as a completely passive service that provides input to Friends group filtering. Given the results of our naïve overlay combination strategy, we will continue to explore alternative integration strategies between RL and LA algorithms.

The assessment of large-scale cyber defense strategies offer numerous challenges, particularly for those who seek to reason about the strengths of competing approaches relative to issues such as the practicality of deployment, cost of operation to resources and management time, impact of countermeasures on non-malicious traffic, the susceptibility to circumvention, applicability to diverse network topologies, and overall effectiveness given the extreme variability in malicious code behavior. The question of assessing the full range of operational characteristics of large-scale cyber defense solutions is extremely difficult, and there are considerable tradeoffs within the limited arsenal of assessment methodologies used today.

Using the simulation parameters identified in this experiment, we intend to expand our work to answer broader questions regarding the comparative assessment of quarantine defense strategies. We shall complete a comprehensive emulation experiment, exploring the dynamics of background traffic and studying the behavioral differences that arise in our simulation and emulation results. We believe differences may arise as a result of the abstractions in our simulations, and understanding these differences may lead to more accurate simulation models.

In several ways, emulation offers a relative middle-ground between the abstraction and low-cost of simulation, and the physical reality and expense of operational testing. Through emulation, researchers can develop repeatable large-scale experiments using applications, operating systems, and in some cases network infrastructure, which are instantiated within virtual process on physical machines. Emulation environments can scale to represent much larger networks than their physical size by instantiating multiple virtual processes per machine. The DETER secure emulation environment [2] offers significant advantage in reducing the cost of creating complex emulation experi-

ments, reducing effort and equipment cost to nearly that of simulation, while offering much of the realism of operational testing. Large-scale worm emulation can provide many of the advantage of simulation including effort, experiment control, and exercise repeatability, while also supporting data capture, background traffic production, and richer component semantics. In addition, the emulated environment can support the execution of real worm code and defense algorithms, providing much deeper insight into the attack/defense dynamics than is currently performed in simulation.

6. Conclusion

We report on the results of an experiment that examines the relative strengths and potential synergies of two complementary worm quarantine strategies: a resource limitation approach and a leap-ahead strategy based on exchanging alert messages among participating gateways in the network. The experiment employs a worm simulation tool, through which we conduct 600 simulations using various worm propagation and scan rates. We explore the potential synergies in combining these two defense strategies, proposing a hybrid combination strategy that merges the two complementary quarantine techniques. We apply the hybrid combination strategy to equivalent simulations and observe performance improvements beyond what either strategy provides independently and a more coherent defense capability over different worm types and speeds. This result may lead to new research directions in hybrid quarantine defenses.

References

[1] K. Anagnostakis, M. Greenwald, S. Ioannidis, A. Keromytis, and D. Li, "A Cooperative Immunization System for an Untrusting Internet*," in Proceedings of the 11th IEEE International Conference on Networks (ICON),* Sydney, Australia, September 2003.

[2] R. Bajcsy and T. Benzel et. al., "Cyber defense technology networking and evaluation," *Communications of the ACM,* Vol 4, No 3, 2004.

[3] L. Briesemeister, P. Lincoln, and P. Porras, "Epidemic Profiles and Defense of Scale-Free Networks," *in Proceedings of the ACM Workshop on Rapid Malcode,* Washington, DC, October 2003.

[4] G. Ganger, G. Economou, and S. Bielski, "Self-Securing Network Interfaces: What, Why, and How," Carnegie Mellon University Technical Report, CMU-CS-02-144, August 2002.

[5] S. Gorman, R. Kulkarni, L. Schintler, and R. Stough, "Least Effort Strategies for Cybersecurity," George Mason University, 2003.

[6] M. Gualtieri and D. Mosse, "Limiting Worms via QoS Degradation," University of Pittsburgh, 2003.

[7] D. Moore, C. Shannon, and J. Brown, "Code Red: A Case Study on the Spread and Victims of an Internet Worm," *in Proceedings of the Internet Measures Workshop,* Marseille, France, November 2002.

[8] D. Moore, C. Shannon, G. Voelker, and S. Savage, "Internet Quarantine: Requirements for Containing Self-Propagating Code," 2003.

[9] D. Nicol, "Models of Active Worm Defense," *in Proceedings of the Measurement, Modeling and Analysis of the Internet (IMA Workshop '04),* Urbana-Champaign, Illinois, January 2004.

[10] D. Nojiri, J. Rowe, and K. Levitt, "Cooperative Response Strategies for Large Scale Attack Mitigation," *in Proceedings of the 3rd DARPA Information Survivability Conference and Exposition,* April 2003.

[11] N. Provos, "A Virtual Honeypot Framework*," in Proceedings of the 12th USENIX Security Symposium,* San Diego, California, August 2004.

[12] S. Staniford, "Containment of Scanning Worms in Enterprise Networks," in *Journal of Computer Security*, 2003.

[13] H. Toyozumi and A. Kara, "Predators: Good Mobile Code Combat against Computer Viruses", *New Security Paradigms Workshop*, Virginia Beach, Virginia, September 2002.

[14] H. Wang, C. Guo, D. Simon, and A. Zugenmaier, "Shield: Vulnerability-Driven Network Filters for Preventing Known Vulnerability Exploits," Microsoft Research, Technical Report MSR-TR-2003-81, February 2004.

[15] N. Weaver, Vern Paxon, S. Staniford, and R. Cunningham, "A Taxonomy of Computer Worms," *in Proceedings of the Workshop on Rapid Malcode*, Washington, DC, October 2003.

[16] M. Williamson, "Throttling Viruses: Restricting Propagation to Defeat Malicious Mobile Code," Hewlett Packard, June 2002.

[17] C. Wong, C. Wang, D. Song, S. Bielski, G.R. Granger, "Dynamic Quarantine of Internet Worms," *in Proceedings of the International Conference on Dependable Systems and Networks (DSN-2004)*, Florence, Italy, June 2004.

[18] C.C. Zou, D. Towsley, and W. Gong, "A Firewall Network System for Worm Defense in Enterprise Networks," University of Massachusetts, Amherst, Technical Report TR-04-CSE-01, February, 2004.