On the Development of an Internetwork-centric Defense for Scanning Worms

Scott E. Coull
Johns Hopkins University
Baltimore, MD 21218
coulls@cs.jhu.edu

Boleslaw K. Szymanski
Rensselaer Polytechnic Institute
Troy, NY 12180
szymansk@cs.rpi.edu

Abstract

The speed with which Internet worms propagate, and their potential for carrying devastating payloads makes them a significant threat to the stability of the Internet. Current approaches for containing these worms are ineffective due to their completely local protection mechanisms – requiring complete deployment for global worm containment. This paper suggests an alternate approach wherein the containment mechanisms are moved within the network itself rather than at end-points. This internetwork-centric approach allows networks within the Internet to not only protect themselves, but also other networks that may not have the containment technology deployed. A novel reputation-based alerting mechanism is used to ensure fair and fast information sharing. The combination of the internetwork-centric containment and reputation-based alerting allows for the creation of an Internet-wide containment mechanism that provides greater protection against fast scanning worms than any previously proposed system, and at the same time providing unequaled resilience to false positives and malicious nodes.

1 Introduction

The phenomenal speed with which Internet worms propagate makes it impossible for manual containment mechanisms to effectively slow or stop the attacks. Several studies have analyzed past worm outbreaks and their propagation methods [7, 6], and point out the inability to significant alter their infection rate without automatic containment. While these worms caused system and network administrators several problems, they are not nearly as destructive or virulent as they could be. The payloads of these worms have been somewhat benign, and the vulnerable population sizes have been relatively small. Staniford, Paxson, and Weaver posit more dangerous worms that could evade even the most sophisticated detection methods and wreak havoc on the Internet in a matter of seconds [13].

Several worm containment systems have been proposed to automatically contain Internet worms [1, 3, 8, 11, 12, 16, 17]. Of these systems, many allow for collaboration among cooperating organizations to preemptively protect enterprise networks from worm attack, while the remaining systems require an attack to occur locally before defensive mechanisms are implemented. While the systems certainly provide various degrees of protection for enterprise networks, there are significant flaws that limit the efficacy of the systems.

For containment systems without collaboration mechanisms, this flaw is manifest in their inability to contain worms that propagate quickly. For collaborative containment systems, the flaw lies in their inability to provide a proper balance between resilience to false alerts and fast reaction to worm outbreaks. The most important flaw with both collaborative and stand-alone containment systems, however, is the need for complete deployment of the systems to all end-point networks in the Internet. Without such a deployment, these systems are unable to make a significant impact on the global propagation of Internet worms. In fact, Wong et. al. have shown that such end-point protection mechanisms are inherently inefficient, and that a network-centric approach to containment provides significant benefits [18].

In this paper, we propose a worm containment system founded on the network-centric containment principles provided by Wong et. al. [18], whereby containment occurs within the core autonomous systems of the Internet topology. The primary benefit of this approach is the ability to have nodes that participate in the containment system protect others that do not have the containment system deployed, simply by filtering both *transiting*, as well as local traffic. It is important to note that Wong et. al. propose the use of network-centric containment in trusted environments, so additional consideration must be given to the specific implementation of the containment in untrusted environments, like the Internet. To accommodate deployment in an untrusted Internet, we provide a novel reputation-based alerting system which uses a weighted voting mechanism to

provide a means for quickly disseminating alerts and acting upon them while remaining resilient to false positives and malicious alerts.

We begin with a discussion of related worm containment research in Section 2. The worm containment system and associated reputation-based alerting systems are described in Section 3. Finally, we provide a simulation system's reaction to an extremely virulent worm in Section 4, discuss our findings in the context of other collaborative approaches in Section 5 and conclude with directions of future research in Section 6.

2 Related Work

There have been a variety of approaches designed to prevent worm infections within a local enterprise network. Methods have been proposed to automatically patch vulnerable hosts [12], throttle scans from infected machines at gateway routers to limit their propagation [17], and automatically reconfiguring local area networks to block communications from infected machines [11, 16]. These local worm containment methods, however, all suffer from an inability to perform preemptive countermeasures due to their purely localized detection and reaction. This causes local containment systems to have a drastically slower reaction time to worm attacks. Additionally, such local countermeasures require an almost complete deployment throughout the networks of the Internet to affect the global propagation of the worm in any meaningful way.

The slow reaction time of localized containment mechanisms has been addressed by introducing mechanisms to share alert status among a number of cooperating organizations. Nojiri et. al. provide a system that dramatically reduces the reaction time of worms by sharing alerts among a predetermined subset of participating nodes [8]. The system proposed by Kannan et. al. instead reports the infection of local hosts to other organizations via implicit alerting with a marker in the header of traffic from infected hosts, or explicitly via alert messages [3]. Both the Nojiri et. al. and Kannan et. al. systems utilize a push model for alerting, and while this model provides excellent reaction to fast scanning worms it can also lead to poor resilience to false positives and malicious alerts.

To overcome the problem with the push model, the COVERAGE system implements a pull mechanism such that an organization periodically queries the alert status of a random subset of cooperating nodes to estimate the approximate virulence of a particular worm outbreak [1]. The cost of this increased resilience to false positives, however, is that the system can not properly protect against fast spreading worms due to the required delay between query periods. Most importantly, the countermeasures for the COVERAGE, Kannan et. al., and Nojiri et. al. systems are implemented to protect only local networks, again requiring significant deployment for effective global worm containment. A discussion of our results in the context of these collaborative systems is given in Section 5.

3 Worm Containment and Collaborative Alerting

As work by Wong et. al. shows, end-point deployment of worm containment systems is inherently inefficient, and is therefore limited in its ability to provide significant global containment of worm outbreaks [18]. A worm containment system where the containment occurs within the core of the network, however, will be able to contain extremely fast worms. Ideally, such containment would occur as close to the infected host as possible, thereby reducing the global impact of the outbreak. Unfortunately, realistic deployment concerns and false alert resilience requirements prevent such a deployment.

Instead, the internetwork-centric containment mechanism described herein aims to provide a scalable implementation which converges toward the optimal blocking scenario as participation and breadth of attack increases. This is achieved by having participating autonomous systems not only block attack traffic destined for local hosts, but also block transiting attack traffic destined for other autonomous systems. The discussion of the internetwork-centric containment system is broken into two distinct parts. First, we describe the mechanisms by which containment of worms occurs within the core of the Internet. Second, we provide the reputation-based alerting mechanism by which we provide resilience to false positives and malicious alerts while ensuring fast reaction to worm outbreaks. Like other approaches, we emphasize that the containment and alerting systems operate independently of detection technologies and can therefore be combined with any number of detection mechanisms. Creation of specific detection mechanisms, however, is beyond the scope of this work.

3.1 Internetwork-centric Containment

When considering the strategy for internetwork worm containment, we must remain cognizant of the unique topology characteristics found within internetwork environments. For instance, the Internet topology has been shown to follow a power-law distribution where a small subset of autonomous systems maintain a large number of connections, while the majority of autonomous systems have few [2]. With this in mind, it seems obvious that to best protect the majority of the Internet from worm attack, we would like the autonomous systems with the largest outdegree within the Internet graph to participate in the containment, thereby filtering the worm from all routes that transit

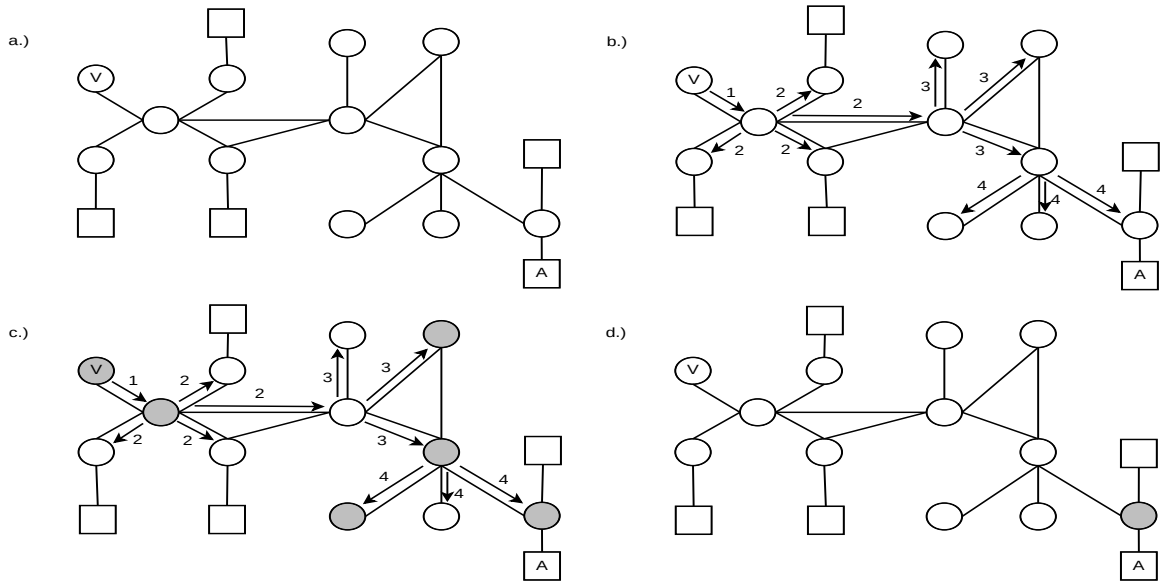

Figure 1. a.) Figure shows participating nodes in the overlay as circles, non-participating nodes as squares. Attacker is denoted as "A" and victim of attack is "V". b.) Propagation of the alert throughout the overlay network. Arrows show direction of propagation while numbers indicate relative timing of alert receipt. c.) Shaded nodes indicate where countermeasures were implemented due to alert increase for "A". d.) After the alerts decay, only the foremost node blocks attack traffic. Other nodes disable countermeasures.

them. Unfortunately, we can not require such participation, so we instead allow for a scalable containment system based on an overlay network among the participants, where adjacencies in the overlay are created through edge contraction between the corresponding autonomous systems within the Internet graph. Significant work has been done to provide methods for creating these overlays, so specific discussion of its creation is omitted [19, 10, 14].

Upon creation of the overlay network, the alerting mechanism utilizes the overlay topology to propagate the alert from the node that detected the worm attack to its direct neighbors, eventually broadcasting it throughout the network. When an alert is received, the node checks whether it has received an alert from the originating victim in the recent past. If it has, then the alert is dropped, otherwise it is applied and propagated. This not only ensures that the same alert is counted only once despite the redundancy that may exist in the overlay topology, but also ensures that a single node can not perform alert flooding to erroneously cause countermeasures to be implemented at other nodes. The definitions of the exact timing of these mechanisms is described with the alerting mechanism in Section 3.2.

Naturally, the node that detected the worm attack would implement filtering countermeasures, such as blocking the offending IP address or their CIDR block, immediately while other nodes in the overlay must reach a certain threshold number of received alerts before implementing their countermeasures. This threshold is discussed in greater detail in the following section. Once a filtering countermeasure is implemented, all traffic that enters the given autonomous system is subject to that filtering policy, including transiting traffic. This ensures that all autonomous systems that would be attacked by routing traffic through the participating system would be protected, as well as the participating autonomous system itself.

Note that the implementation of countermeasures within the containment system directly corresponds to the breadth of attacks. Therefore, if the attacks are localized to a specific portion of the containment overlay, then the filtering will also be localized. Once attacks from a single host become prevalent enough, we can see that a type of perimeter of blocking would be implemented, which would slowly converge towards the attacker's autonomous system. This perimeter, of course, is limited by the placement of the participating autonomous systems within the Internet topology, but converges toward an optimal blocking strategy as participation in the containment system increases.

We further optimize the blocking perimeter by allowing autonomous systems within the overlay to remove their blocking countermeasure when no additional traffic from the offending host is detected. The autonomous systems closest to the attacker would continue filtering traffic be-

cause it is the so-called first line of defense and is therefore the first participating node to receive the attack traffic. Nodes farther from the attacker will not see the traffic because it is being dropped by the first line of defense, thereby allowing them to remove their filtering countermeasures for that attacker and free resources. This naturally ensures that for each attacker the global countermeasure strategy becomes optimal as the breadth of the attack increases, requiring only the minimum of participating nodes to perform the filtering countermeasures.

An example of the containment system reacting to an alert is given in Fig. 1. In this situation, we see the scenario where a single infected host, A, triggers an alert from the victim of the attack, V. The broadcast of the alert message occurs throughout the overlay network among the participating nodes. Since some of those participating nodes have several previous alerts for A, this latest alert forces them to enact countermeasures and block traffic from A. Over time, the reputation decays and nodes are able to remove their countermeasures, except for the foremost node which continues to see attack traffic from A. That foremost node retains its countermeasures for as long as the attack traffic is detected, and therefore provides protection from A for the entire overlay network and other nodes that do not participate in the containment system. Though this scenario deals with only a single attacker and victim for clarity, this same procedure occurs with every alert generated in a concurrent fashion. Also, remember that the overlay may in fact have many autonomous systems between neighboring nodes in the overlay network, and that those nodes would also be protected even though they are not explicitly depicted in the figure.

3.2 Reputation-based Alerting

Without proper safeguards, a containment system such as the one proposed here can easily become a devastating weapon for attackers. Not only should the alerting mechanism prevent the abuse of the containment system for malicious purposes, but it should also provide fair and fast alerting capabilities that ensure worms are contained in the most efficient way possible. One intuitive notion for implementing alerting mechanisms is the idea of reputation. Humans often use recommendations gained from friends and colleagues to form opinions of various goods, services, and people without prior experience.

We leverage a similar concept to provide an alerting mechanism for the internetwork-centric containment system. This alert system allows every participating autonomous system to retain an *independent* reputation value for malicious hosts which can be increased when alerts are received indicating malicious behaviors. Hence, this reputation value in fact indicates the level of disrepute for any given host. Once this reputation value increases above a particular threshold at any of the participating autonomous systems, the malicious host's traffic is blocked at that participating system.

Moreover, just as humans transitively trust acquaintances less than direct relations (i.e. trust close friends more than friends of friends), the alert system utilizes the overlay network as a type of social network among autonomous systems. The intuition behind this definition of trust is based on the probability that a given node routes traffic for the local network. It follows that nearby nodes within the Internet graph necessarily route a larger proportion of the traffic for the local network than more distant nodes. Therefore, nodes that are neighbors within the overlay necessary have close associations within the full Internet graph due to the edge contraction, and trust those neighboring autonomous system's alerts more than distant autonomous systems.

Finally, the effects of these recommendations tend to diminish over time as new information or direct behavior is observed. The use of reputation in similar social network scenarios has been shown to be an excellent way to quickly propagate information [4]. We formalize these notions of reputation as one equation and two inequalities, given as (1)-(3), and provide necessary definitions in Table 1.

The system is initialized at each participating node with the reputations for all hosts set to zero. When an alert is received, (1) is implemented to increase the value of the attacker's reputation based upon the distance the alert has traveled in the overlay network, and the number of neighbors the current node has. Clearly, the strength of an alert is multiplicatively reduced based on the distance the alert has traveled before reaching the current node. This limits the effective range of the alert, and creates the localized containment effects that we have previous described.

When the distance from the alert originator is zero (i.e. the current node originated the alert), the value of the reputation is increased by one. Also, when the distance from the alert originator is one (i.e. the originator is a direct neighbor), the reputation is increased in such a way that when a majority of the direct neighbors provide alerts, the sum of their alert values will equal one. Thus, we can set the threshold for countermeasure implementation to one, as it allows locally generated alerts to immediately implement countermeasures while requiring the equivalent of a weighted majority vote of remote alerts.

Additionally, (3) stops the alert propagation once the perceived strength of the alert has reached a level where it will be of no use to other nodes. The distance that an alert has propagated thus far must be less than the number of remaining neighbors to propagate the message to. If it is not, alert propagation ceases. The use of both (1) and (3) limits the global effects of any possible malicious use of the system while allowing for a convergence to optimal global contain-

Table 1. Definition of reputation variables

Reputation Variable	Definition
a	Attacker
t	Current time
λ	Decay period
m	Management period
N	Number of neighbors
rep(x,y)	Reputation value for attacker x at time y
d(z)	Distance from current node to alert originator
v(x,y)	Latest alert originator to send alert for attacker x before time y
p(x,y)	Time of previous alert about attacker x before time y

$$rep(a,t) = rep(a,p(a,t)) + \frac{1}{\left(\lfloor \frac{N+1}{2} \rfloor d(v(a,t))\right) + 1} \tag{1}$$

$$rep(a,t) \geq \left(1 + \frac{1}{\lfloor \frac{N+1}{2} \rfloor}\right)^{\lfloor \frac{t}{m} \rfloor (\frac{m}{\lambda})} \tag{2}$$

$$d(v(a,t)) \leq (N-1) \tag{3}$$

ment from broad worm attacks.

The decay of the stored reputations at each participating node occurs by reducing the reputation at discrete intervals, known as management periods. These management periods are defined on a per-node basis and it indicates the time when the evaluation of the decay and the changes in the reputation value actually occur. Note that this is different than the actual length of the effective decay, known as the λ period. In simpler terms, the λ period is the decay interval where as the management period is the interval at which that decay is actually applied and recorded. At each management period, the reputation is reduced multiplicatively by a factor of:

$$\left(\frac{\lfloor \frac{N+1}{2} \rfloor}{\lfloor \frac{N+1}{2} \rfloor + 1}\right)^{\frac{m}{\lambda}}$$

Thus, when the reputation value is reduced to a value less than one, the countermeasures are stopped. The countermeasure implementation equation can be succinctly derived by combining the required threshold for countermeasure implementation with the decay rate, as provided in (2).

The reputation-based alerting system ensures that only one alert about a given attack from a particular victim is applied every management period. Not only does this prevent duplicate alerts, but it also has the added benefit of preventing alert flooding. In the case of alert flooding, a specific node would attempt to send multiple alerts for a single attacker, thereby attempting to erroneously enact a block on them at other nodes in the network. Since only one alert is allowed per management period, there would need to be significant collaboration among nodes to cause erroneous blocking.

4 Evaluation

To test the containment abilities and the false positive resilience of the internetwork-centric containment system, we employ a custom discrete event simulator. The simulation uses a scale down factor of $\frac{1}{2}$ to model the autonomous system distribution of the Internet as derived from the Route Views project [9]. Thus, we use 9,000 autonomous systems connected in a power-law topology as created by the BRITE topology generator [5]. This ensures that the topology provided will approximately model the distribution of connections among autonomous systems within the Internet graph. We also assume an address space of 2^{31} uniformly distributed among all autonomous systems. Our simulated worm has a 1% vulnerable population, which is also uniformly distributed among all 9,000 autonomous systems in the simulation. While the vulnerable population distribution is not completely consistent with observations of some worms, this uniform distribution represents a worst case scenario in which the alerting mechanism will have a minimum density of alerts per unit of time in any particular area of the topology. Therefore, such a scenario provides a lower bound for the performance of our containment system and alerting mechanism.

The simulated worm has a scanning rate similar to the observed scanning rates of the Slammer worm [6]. As studies of the Slammer worm have shown, however, shared access links limit the effective scanning rate to approximately

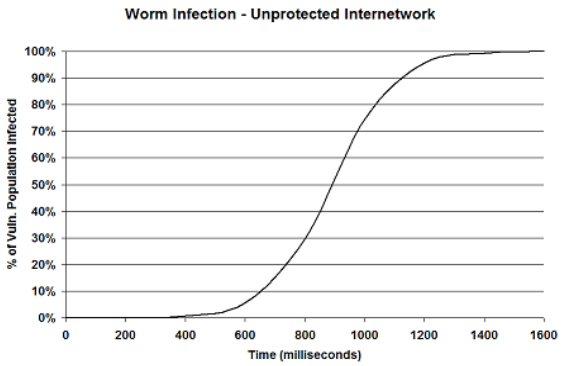

Figure 2. Infection rate of the simulated worm within the internetwork with no protection

4,300 scans per second for each shared access link, and therefore this limit must be modeled properly in the scaled down simulation [15]. For this simulation, we assume that the limiting access link occurs within the autonomous systems, thereby limiting the total scanning rate for all infected hosts within that autonomous system to 4,300 scans per second. Since the simulated worm has a scanning rate equal to that of Slammer and a vulnerable population that is many orders of magnitude larger than any worm to date, this simulated worm is far more virulent than any worm released, and provides a lower bound on the performance of our system in containing worm outbreaks.

Due to the fast propagation of the worm, the λ and management settings can be set fairly low. The reputation system's management and λ periods are, therefore, set to ten seconds for all tests. Additionally, we assume that once a single infection occurs within an autonomous system, all vulnerable hosts in the autonomous system are infected, and these infections are not cleaned during the simulation.

We perform a number of tests to assess the performance of the system in a partial deployment scenario. To minimize the stochastic effects of the worm's scanning on the results, we took five trials of each test over one hour of simulated time. The geometric mean is used to provide the central behavior of the system over all trials. To provide a baseline for the virulence of the worm, we performed a simulation of the spread of the worm in a totally unprotected environment, as shown in Fig. 2. The worm infects all vulnerable hosts in approximately 1.6 seconds, thereby supporting claims that this simulated worm far outstrips all known worms in terms of virulence and capabilities for fast propagation.

Tests were performed to measure the percentage of the vulnerable population infected by the worm for a partial containment system deployment of 50% and 25% of all autonomous systems. These partial deployments contained a random subset of nodes within the generated topology, connected by the overlay network previously described. In particular, these tests attempt to quantify the level of participation required to stop the extremely virulent worm described above, as well as the overall effectiveness of reputation-based alerting and internetwork-centric containment.

Additionally, tests were performed to model the effects of alert loss on the containment system's performance. In these tests, an alert can be dropped and fail to send at any point during the propagation, including at the originating victim. Finally, we also tested the effects of colluding malicious systems and ambient false positives on the system with a 50% participation. Colluding malicious systems were simulated by labeling a percentage of the participating autonomous systems as malicious. Each of these malicious systems produce an alert about the same autonomous system within a single management period. Ambient false positives were simulated by having a random subset of the participating hosts produce alerts for random autonomous systems. In both cases, we measure the number of autonomous systems that begin blocking due to reputation increase.

4.1 Results

Fig. 3 shows the containment abilities of the internetwork-centric system for participation levels of 50% and 25%, respectively. As you can see, the system is able to efficiently contain the worm with a participation level of 50%. The 25% participation level, however, is not able to contain the worm, though it does slow the worm's propagation dramatically. While this level of deployment does not provide complete global containment, the system still dramatically slows the progress of the worm. It is important to point out, however, that the worm is extremely virulent with a 1% vulnerable populations and fast propagation. At such a high vulnerable population, the worm is difficult to contain because nearly 430 scans each second of simulated time find and infect a new host. While 1,075 of the total scans for a single worm are detected by participating autonomous systems on average every second, the growth of the worm soon outstrips the protection of the containment system. Only the first of those alerts are usable due to the limitations imposed by the reputation-based alert system in Section 3.2 based on the length of the management period. The slow down in the propagation is significant, and some non-participating autonomous systems remain protected after one hour of time.

To test the performance of the internetwork-centric containment strategy with dropped alert messages, we set the simulation to drop alerts at any point during their propagation with 1, 5, and 10% probability. The only node that is guaranteed to increment the reputations of the attackers is the victim. As you can see by Fig. 4, the percentage of alerts that are dropped does not adversely affect the per-

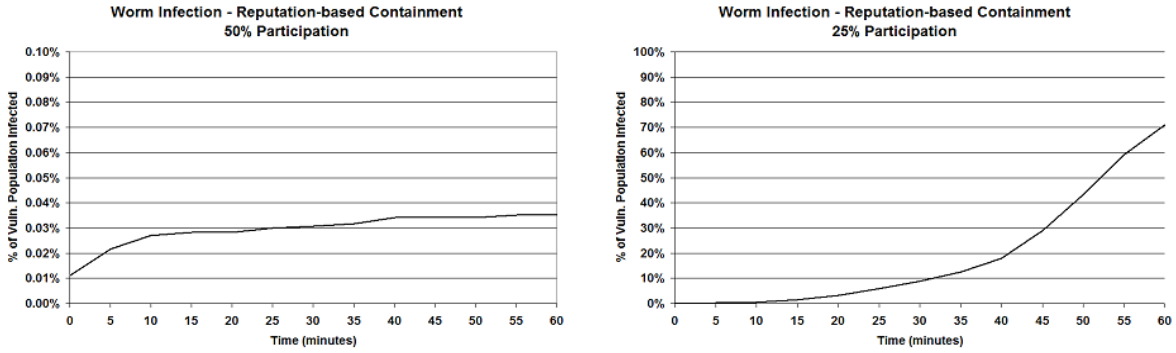

Figure 3. Infection rate of simulated worm with 50 and 25% of nodes in the internetwork participating in containment

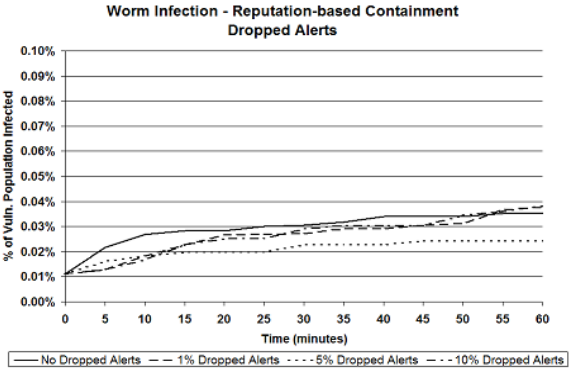

Figure 4. Infection rate of the simulated worm with 50% of nodes in the internetwork participating in the reputation system with varying alert propagation reliability

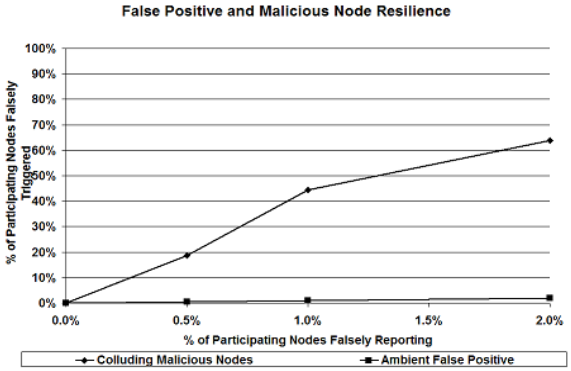

Figure 5. Percentage of falsely triggered nodes versus percentage of nodes reporting false alarms

formance of the containment. The chart clearly shows the stochastic effects of the worm's propagation, but there is no clear impact from the dropped alerts. The reason for the limited effect of dropped alerts is the internetwork-centric containment itself. With the dropped alerts, the only node that is guaranteed to increase its reputation value is the originating victim. The results indicate that the majority of the blocking task is performed initially by the victim node before the system converges to optimized containment. This lends credibility to the usefulness of internetwork-centric containment as a paradigm for virulent worm containment.

The most important feature of the reputation-based alert system is its resilience to false positives and malicious nodes. As you can see in Fig. 5, the scenario with malicious collusion causes a large number of autonomous systems to falsely block traffic. While this is certainly not desirable, it is quite difficult to provide fast reaction to widespread worm traffic while providing this kind of resilience to collaborating nodes. This result is fairly intuitive, as the malicious systems are actually acting exactly as our system intends. Such an attack is difficult to combat without crippling the reaction time of the system by requiring more corroboration of alerts. Interestingly, our reputation-based alerting system outperforms all other collaborative systems against simple false positive alerts. The increase in false positives is linear with the growth of the false reporting population, as only those falsely reporting node perform the blocking. The resilience to false positive alerts is quite intuitive due to the reputation-based alert mechanism, as significant corroboration is required for countermeasures to be enacted.

Table 2. Comparison of the lower bound results for collaborative containment systems

System	% Vuln. Population Infected	% Nodes Falsely Alerted
Nojiri et. al. [8]	20%	75%
Kannan et. al. [3]	34%	1%
COVERAGE et. al. [1]	20%	40%
Internetwork-centric	0.004%	2%

5 Discussion

To provide context for the results provided, we discuss the results of three state-of-the-art collaborative containment systems. The Nojiri et. al. and Kannan et. al. systems, as described in Section 2, implement a localized containment strategy with a push model for alerting cooperating nodes [8, 3]. The COVERAGE system also implements localized containment, but uses a polling mechanism to share alert information among participants. These previously proposed systems will be compared to the results above in terms of their effects on worm propagation and their resilience to false positives. In all cases, the worms used to test these systems are far less virulent than the one described in Section 4 used to test our internetwork-centric containment system.

To test their system, Nojiri et. al. create a simulation with 729 networks, all of which participate in the containment, and 5832 vulnerable hosts uniformly distributed among all networks [8]. Specific parameters for the scanning rate are not given, but the authors specify that one host is infected per scanning period. Thus, this simulated worm is extremely slow and the containment system has been completely deployed in all networks within the simulation. Even with these somewhat unrealistic settings, the system allows approximately 1200 hosts (20%) of the vulnerable population to become infected before the worm is stopped. In addition, the authors found that with only 36 of the participating networks (5%) providing false positives a total of approximately 550 networks (75%) implemented countermeasures.

Kannan et. al. simulate the propagation of a worm with a scanning rate of 20,000 scans per second and a vulnerable population of 0.05% [3]. These vulnerable hosts were uniformly distributed among 10,000 networks where each network has deployed the worm containment mechanism. With the optimal settings, Kannan et. al. contain the worm to infecting only 0.0005% of the vulnerable population. However, their scheme relies on a static setting for determining the number of alerts to receive before implementing countermeasures. When this alert setting provides resilience to 1% of the participating nodes providing false positives, their performance drops by allowing more than 30% of the vulnerable population to become infected.

Finally, COVERAGE creates a network of 2,000 domains with 800,000 hosts uniformly distributed among all domains, all of which have the containment system deployed [1]. Again, COVERAGE does not specify a scanning rate but instead varies the rate of infection attempts *per minute*. With only 0.6 infection attempts per minute, the COVERAGE system is able to reduce the worm to only infecting approximately 0.005% of the vulnerable population. As the number of infection attempts per minute increases to 2, the COVERAGE system quickly drops in performance allowing nearly 20% of the vulnerable population to be infected. Additionally, when 2% of the participants are providing false alerts, the COVERAGE system only triggers countermeasures at 40% of the nodes.

Each of the collaborative worm containment systems, therefore, have implemented a worm that is far less virulent than the one used in our evaluation in Section 4 due to our large vulnerable population size. Moreover, each of these systems is evaluated with complete deployment to all nodes in the simulated environment whereas our simulation assumes at most a 50% deployment of the system randomly throughout the internetwork. Thus, our evaluation uses weaker assumptions on the environment within which the containment system will be deployed.

Though our assumptions on the deployment environment are weaker, our internetwork-centric approach still exceeds the performance of all of these systems in terms of containment and false positive resilience. For containment performance in the case of 50% deployment, our system allows a maximum of 0.004% of the vulnerable population to be infected while allowing at most a linear number of false countermeasure implementations with the number of falsely reporting nodes.

In contrast, only the Kannan et. al. system performs closely with regard to containment. However, when the Kannan et. al. system is set to provide a similar level of false positive resilience, its performance falls drastically below that of our system. Clearly, the internetwork-centric containment using reputation-based alerting provides the best solution to automatically containing fast scanning worms. A table summarizing lower bound performance of each collaborative containment system is given as Table 2. This table compares the worst-case performance reported for each of the containment systems in terms of its ability to prevent the infection of the vulnerable population, and its ability to prevent nodes from falsely implementing countermeasures.

6 Conclusion

Worm containment remains a difficult task that will only become more difficult as worm propagation mechanisms evolve and become more efficient. Though many systems have been proposed for the containment of worms, none allow for complete global containment under partial deployment due to the localized nature of their containment mechanisms. In addition, collaborative containment systems used to share alerts among untrusted parties provide a poor balance between reaction to fast spreading worms and resilience to false positives and malicious alerts.

The internetwork-centric containment system proposed here is able to provide a fair alert thresholding system with the use of a novel reputation mechanism while allowing for efficient containment of worms due to its internetwork-centric approach to containment. The use of these novel tools in worm containment allow the internetwork-centric containment system to effectively quarantine an extremely virulent worm under partial deployment scenarios, while offering significant improvements in resilience to false positives and colluding malicious nodes. We have shown that not only are we able to provide better containment than other collaborative containment systems under more realistic assumptions, but that we also provide better resilience to false positives.

Moreover, the design of this internetwork-centric containment system allows for extensible implementation against other one-to-many attacks, such as e-mail viruses and spam. Alert propagation and containment occurs just as if a worm outbreak were occurring, but the associated λ and management periods would be changed to better accommodate the propagation times for these specific types of malicious behavior. Just as the internetwork-centric containment provides a convergence to optimal containment for worm traffic, the system would also provide containment for pervasive spammers or wide-spread mass mail viruses. This indicates the most significant benefit of the internetwork-centric containment, which is its ability to provide an internetwork, such as the Internet, with an autonomic defense mechanism to prevent malicious systems from causing devastation throughout the Internet. Such an extensible defense mechanism has not yet been proposed, and this paper provides an initial study of its feasibility and capabilities, indicating significant promise in its applicability to preventing a wide range of malicious behaviors.

Acknowledgments

Research of Boleslaw Szymanski is continuing through participation in the International Technology Alliance sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the U.S. Government, the UK Ministry of Defense, or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

The authors would like to express their sincere thanks to Charles Wright and our anonymous reviewers for their insightful comments.

References

[1] K. Anagnostakis, M. Greenwald, S. Ioannidis, A. Keromytis, and D. Li. A Cooperative Immunization System for an Untrusting Internet. In 11^{th} *International Conference on Networks*, 2003.

[2] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On Power-Law Relationships of the Internet Topology. In *ACM SIGCOMM*, pages 251–262, 1999.

[3] J. Kannan, L. Subramanian, I. Stoica, and R. Katz. Analyzing Cooperative Containment of Fast Scanning Worms. In *USENIX SRUITI Workshop*, August 2005.

[4] H. Kautz, B. Selman, and M. Shah. Referral web: combining social networks and collaborative filtering. *Commun. ACM*, 40(3):63–65, 1997.

[5] A. Medina, A. Lakhina, I. Matta, and J. Byers. BRITE: An Approach to Universal Topology Generation. In *International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems*, August 2001.

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

[7] D. Moore, C. Shannon, and K. Claffy. Code-Red: A Case Study on the Spread and Victims of an Internet Worm. In 2^{nd} *ACM SIGCOMM Workshop on Internet Measurement*, pages 273–284, 2002.

[8] D. Nojiri, J. Rowe, and K. Levitt. Cooperative Response Strategies for Large Scale Attack Mitigation. In *DARPA Information Survivability Conference and Exposition*, pages 293–302, April 2005.

[9] Route Views. `http://www.routeviews.org`.

[10] A. Rowstron and P. Druschel. Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer Systems. In *IFIP/ACM International Conference on Distributed Systems Platforms*, pages 329–250, November 2001.

[11] R. Scandariato and J. C. Knight. The Design and Evaluation of a Defense for Internet Worms. In 23^{rd} *IEEE International Symposium on Reliable Distributed Systems*, pages 164–173, 2004.

[12] S. Sidiroglou and A. Keryomytis. Countering Network Worms Through Automatic Patch Generation. *IEEE Security and Privacy*, 3(6):41–49, 2005.

[13] S. Staniford, V. Paxson, and N. Weaver. How to 0wn the Internet in Your Spare Time. In 11^{th} *USENIX Security Symposium*, 2002.

[14] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balikrishnan. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. In *ACM SIGCOMM 2001*, August 2001.

[15] N. Weaver, I. Hamadeh, G. Kesidis, and V. Paxson. Preliminary Results Using Scale-down to Explore Worm Dynamics. In 2^{nd} *ACM Workshop on Rapid Malcode*, October 2004.

[16] N. Weaver, S. Staniford, and V. Paxson. Very Fast Containment of Scanning Worms. In 13^{th} *USENIX Security Symposium*, pages 29–44, 2004.

[17] M. Williamson. Throttling Viruses: Restricting Propagation to Defeat Malicious Mobile Code. In 18^{th} *Annual Computer Security Applications Conference*, pages 61–68, 2002.

[18] C. Wong, C. Wang, D. Song, S. Bielski, and G. Ganger. Dynamic Quarantine of Internet Worms. In *International Conference on Dependable Systems and Networks*, June 2004.

[19] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz. Tapestry: A Resilient Global-Scale Overlay for Service Deployment. *IEEE Journal on Selected Areas in Communications*, 22(1):41–53, 2004.