Worm Versus Alert: Who Wins in a Battle for Control of a Large-Scale Network?

James Aspnes[1], Navin Rustagi[2], and Jared Saia[2]

[1] Department of Computer Science, Yale University email: aspnes@cs.yale.edu. This research was partially supported by NSF CNS 0435201

[2] Department of Computer Science, University of New Mexico, Albuquerque, NM 87131-1386; email: {navin, saia}@cs.unm.edu. This research was partially supported by NSF CAREER Award 0644058 and NSF CCR-0313160

Abstract. Consider the following game between a worm and an alert[3] over a network of n nodes. Initially, no nodes are infected or alerted and each node in the network is a special *detector* node independently with small but constant probability. The game starts with a single node becoming infected. In every round thereafter, every infected node sends out a constant number of worms to other nodes in the population, and every alerted node sends out a constant number of alerts. Nodes in the network change state according to the following four rules: 1) If a worm is received by a node that is not a detector and is not alerted, that node becomes infected; 2) If a worm is received by a node that is a detector, that node becomes alerted; 3) If an alert is received by a node that is not infected, that node becomes alerted; 4) If a worm or an alert is received by a node that is already infected or already alerted, then there is no change in the state of that node.

We make two assumptions about this game. First, that an infected node can send worm messages to any other node in the network but, in contrast, an alerted node can send alert messages only through a previously determined, constant degree overlay network. Second, we assume that the infected nodes are intelligent, coordinated and essentially omniscient. In other words, the infected nodes know everything except for which nodes are detectors and the alerted nodes' random coin flips i.e. they know the topology of the overlay network used by the alerts; which nodes are alerted and which are infected at any time; where alerts and worms are being sent; the overall strategy used by the alerted nodes; etc. The alerted nodes are assumed to know nothing about which other nodes are infected or alerted, where alerts or worms are being sent, or the strategy used by the infected nodes.

Is there a strategy for the alerted nodes that ensures only a vanishingly small fraction of the nodes become infected, no matter what strategy is used by the infected nodes? Surprisingly, the answer is yes. In particular, we prove that a simple strategy achieves this result with probability approaching 1 provided that the overlay network has good node expansion. Specifically, this result holds if $d \geq \alpha$ and $\frac{\alpha}{\beta(1-\gamma)} > \frac{2d}{c}$, where α and

[3] Specifically, we consider self-certifying alerts[6], which contain short proofs that a security flaw exists and thereby eliminate false alerts.

β represent the rate of the spread of the alert and worm respectively; γ is the probability that a node is a detector node; d is the degree of the overlay network; and c is the node expansion of the overlay network. Next, we give empirical results that suggest that our algorithms for the alert may be useful in current large-scale networks. Finally, we show that if the overlay network has poor expansion, in particular if $(1-\gamma)\beta > d$, then the worm will likely infect almost all of the non-detector nodes.

Keywords: Self-certifying alert, worm, overlay network, peer-to-peer, expander graphs, epidemic processes
Contact Info: Email: saia@cs.unm.edu, Fax: (505) 277-6927

1 Introduction

Attacks on the Internet are characterized by several alarming trends: (i) increases in frequency: large-scale attacks are approximately doubling every year [22]; (ii) increases in speed: the recent slammer worm infected 90% of vulnerable hosts within 10 minutes [17]; and (iii) increases in severity: the slammer worm had many unforeseen consequences including failures of 911 emergency data-entry terminals, network outages, and canceled airline flights, [17, 7, 11, 10]. In addition, there has been a broadening of motivations for attack to include extortion [23, 1]; phishing [8, 24, 12]; sending anonymous spam [15, 13]; and political reasons [18, 19]. Modern computer worms simply propagate too quickly for human detection. Since attacks are now occurring at a speed which prevents direct human intervention, there is a need to develop automated defenses. Since the financial, social and political stakes are so high, we need defenses which are *provably good* against a worst case attacks.

A promising recent result in this direction is the development of self certifying alerts(SCAs)[6]. An SCA is a short, machine verifiable, automatically generated proof that a security flaw exists. Because an SCA is short, it is easily propagated through a network. Because an SCA is efficiently verifiable, false positives are eliminated. SCAs are generated by dedicated machines called detectors. Detectors run instrumented software to automatically detect a worm, determine which vulnerability the worm exploits, and then generate an SCA for the worm, i.e. a short proof that the vulnerability the worm exploits does in fact exist. After receiving and verifying an SCA, a machine can generate a filter that blocks infection by analyzing the exploit which the SCA proves exists. Because the SCA focuses on the security flaw exploited by a worm, rather than the textual content of the worm, SCAs can easily be created for polymorphic worms. Recent empirical results suggest that SCAs can be generated, checked and deployed efficiently. For example, the Vigilante system [5] takes 18 milliseconds to generate an SCA for the Slammer worm, the resulting SCA is 457 bytes long, the time to verify this SCA is 10 milliseconds, and the time to create a filter from the verified SCA is 24 milliseconds. These times for SCA generation, verification and filter creation are on the same scale as the time it takes a worm to infect a machine. Vigilante performs similarly for two other Internet worms, Code Red and Blaster.

Distribution of alerts in the Vigilante system is performed by the Pastry[20] peer-to-peer overlay network. It is shown empirically that a very small fraction of special detector nodes is enough to ensure that a worm infects no more than 5% of the vulnerable population. While these initial results are promising, several critical problems remain. First, Vigilante requires that the nodes participating in the overlay network all be resistant to infection. Second, Vigilante requires that the topology of the overlay network be hidden from the worm. These two assumptions may hold true for an overlay network owned and operated by a single company, but seem unlikely to hold for a large-scale open source peer-to-peer network. Finally, while the Vigilante systems performs well empirically against currently known worms, the system has no known theoretical guarantees against all worms. In this paper, we focus exclusively on the problem of distribution of alerts through an overlay network and address these three problems.

1.1 Our Model

We model our problem of alert distribution as a game between a worm and an alert over a synchronous network. Initially, no nodes are infected or alerted and each node in the network is a special *detector* node independently with fixed probability γ. The game starts with a single node becoming infected. In every round thereafter, every infected node sends out β worms to other nodes in the population, and every alerted node sends out α alerts for fixed constants α and β. Nodes in the network change state according to the following four rules: 1) If a worm is received by a node that is not a detector and is not alerted, that node becomes infected; 2) If a worm is received by a node that is a detector, it is not infected, instead it becomes alerted; 3) If an alert is received by a node that is not infected, that node becomes alerted; 4) If a worm or an alert is received by a node that is already infected or already alerted, then there is no change in the state of that node.

We make two assumptions about this game. First, an infected node can send worm messages to any other node in the network but, in contrast, an alerted node can send alert messages only through a previously determined, constant degree overlay network. In other words, the alert-spreading algorithm is “polite” in the sense that it does not bombard arbitrary nodes with alerts unless it knows that they are interested in receiving them. Since the worm is not required to be polite, it is not constrained by the overlay network, although a particularly sophisticated worm may exploit the structure of the overlay network for its own purposes. An edge in this overlay network represents an agreement between two nodes to accept SCAs from each other. Second, we assume that the infected nodes are intelligent, coordinated and essentially omniscient. In other words, the infected nodes know everything except for which nodes are detectors and the alerted nodes’ random coin flips i.e. they know the topology of the overlay network used by the alerts; which nodes are alerted and which are infected at any time; where alerts and worms are being sent; the overall strategy used by the alerted nodes; etc. Moreover, the worm is unconstrained in which nodes it attacks. For example, it could always try to infect nodes which have never been infected before. The alerted nodes are assumed to know nothing about which other nodes are infected or alerted, where alerts or worms are being sent, or the strategy used by the infected nodes. Also the number of messages an alerted node can send is constrained by the degree of the graph.

1.2 Results

In our results, we make use of a d-regular overlay network with node expansion c. As a concrete example, a random d-regular graph has node expansion $c = d/5-1$ with high probability[4]. Throughout this paper, we use the phrase with high probability (w.h.p) to mean with probability at least $1 - 1/n^{\epsilon}$ for some fixed $\epsilon > 0$. Let RANDOM be the algorithm that has each alerted node in each round send out alerts to α nodes selected uniformly at random without replacement from its neighbors in the overlay. Our main theoretical results are stated below as the following two theorems which are proven in Sections 2 and 4 respectively.

Theorem 3: If $d \geq \alpha$ and $\frac{\alpha}{\beta(1-\gamma)} > \frac{2d}{c}$, then the algorithm RANDOM ensures that, w.h.p, only $o(n)$ nodes are ever infected.

Theorem 6: If the overlay network has bounded degree d and $\beta(1-\gamma) > d$, then any alert algorithm in expectation will save a fraction of non-detector nodes that approaches 0 as n gets large

Our empirical results, presented in Section 3, show that if the overlay network is a d-regular random graph, as n grows large, the algorithm RANDOM saves an increasingly large fraction of the nodes against a worm that spreads uniformly at random. For example, for $n = 10^6$, $d = 100$, $\beta = 1$, $\alpha = 5$ and $\gamma = .02$, we were able to save 99% of the nodes on average.

1.3 Other Related Work

Several approaches for generating self-certifying alerts have been proposed recently (see e.g. [9, 14, 3], but few systems have been proposed for disseminating those alerts. The Vigilante system and its limitations have been discussed above. Zhou et al. [16] propose a system for distributing alerts over a network, but their system is focused on confronting worms that can spread only through the same overlay network through which the alert is spreading. Vojnovic and Ganesh [25] and Shakkottai and Srikant [21] perform exhaustive analytical and empirical studies of the effectiveness of different types of alert dissemination. However, their work focuses only on worms that spread uniformly at random in the network. In contrast, our work considers worms that may use smarter dissemination strategies.

[4] see [4] for an algorithm for sampling from random d-regular overlay networks in a distributed manner

2 Alert versus worm in an expanding overlay network

In this section, we focus on d-regular graphs for our overlay network. We show that for a suitable choice of parameters and a particular type of overlay network, we are able to save most of the nodes from getting infected with high probability. More precisely, at the end of the process only $o(n)$ nodes get infected, and all other nodes get alerted.

The essential idea is that we want the long-run growth rate of the set of alerted nodes to be higher than the rate for the infected nodes. The rate for infected nodes is easy to calculate; assuming an optimal choice of targets, each infected node infects on average an additional $\beta(1-\gamma)$ nodes per round. The rate for alerted nodes is trickier, as alerted nodes are limited by the structure of the overlay network. But we can get a lower bound on the expected rate during the early parts of the protocol by observing that A alerted nodes will between them have at most dA neighbors, of which at least cA will not already be alerted, where c is the expansion parameter of the network. It follows that each alerted node will attempt to alert on average at least $\alpha(c/d)$ unalerted nodes at each step. In the absence of the worm, this would give the growth rate of the alerted nodes; with M infected nodes, we must subtract these from the pool of new alerted nodes (using the simplifying assumption that the worm successfully concentrates itself on the boundary of the set A). Fortunately these lost infected nodes are compensated for somewhat by the boost of $\gamma\beta M$ new alerted nodes from triggered detectors.

This overview ignores two important details. Because we want a high-probability bound, it is not enough simply to consider expected growth rates. And because the expansion factor applies only for sets with $n/2$ or fewer elements, we must consider separately the case where the set of alerted nodes is larger. We handle both problems by dividing the execution into three phases. Phase I starts with a single infected node and ends when $\ln n$ worm messages have been received by nodes in the network. During this phase we ignore the spread of alerts and content ourselves with getting only the $\Theta(\gamma \ln n)$ alerted nodes that result from successful detections. Phase II starts at the end of of Phase I. During this phase we use the fact that the number of infected and alerted nodes are both $\Omega(\log n)$ to show that both the worm and the SCA propagate at close to the expected rate with high probability; the key point is that when the populations of both are large enough, Chernoff bounds apply to the increases. Phase II ends when n/d^2 nodes have been alerted by the SCA; at this point we can no longer rely on the expansion properties of the network and must resort to a different analysis. Note that there are expansion properties till the end of Phase II. For this analysis, done in Section 2.3, we show that in constant number of steps, we would alert n/2 nodes and then after c log(log(n)) further steps we would have only o(n) not alerted or not infected nodes. Thus we would have shown that only o(n) nodes could have been infected and $\theta(n)$ nodes have been alerted.

In the remainder of this section, all lemmas that bound a random variable's value for t rounds hold with probability greater than or equal to $1 - t/n^c$ for some fixed constant $c > 0$. Also for all the remaining lemma's in this section, $d \geq \alpha$.

2.1 Phase I

Let Z be the set of nodes that receive the first $\ln n$ worm messages; i.e., the set of nodes that receive worm messages in Phase I.

We write A_i for the number of nodes alerted at time t, counting from the end of Phase I; thus A_0 is the number of nodes alerted in Z.

Lemma 1. *At the end of Phase I, (a) the expected number of alerted nodes* $\mathrm{E}[A_0]$ *is at least* $\gamma \ln n$*; and (b) for any* $c > 0$*, there exists a constant* $\delta \leq 1/2$*, such that with probability greater than* $1 - 1/n^c$*,* $(1 - \delta)\,\mathrm{E}[A_0] \leq A_0$

Proof. For each $v \in Z$, let X_v be the indicator random variable for the event that v is alerted in Phase I and let Y_v be the event that v is a detector node. While the X_v are not necessarily independent, we do have that $X_v \geq Y_v$ for all v, and thus $A_0 = \sum_{v \in Z} X_v \geq \sum_{v \in Z} Y_v$. It follows that $\mathrm{E}[A_0] \geq \sum \mathrm{E}[Y_v] = \gamma |Z| = \gamma \ln n$. The second part is an immediate application of Chernoff bounds. ■

It follows that A_0 is $\Theta(\ln n)$ with high probability.

2.2 Analysis of Phase II

For the second phase, begin by comparing the number of infected nodes in the actual process with the number of infected nodes in an infinite graph where the SCA has no effect on the spread of the worm. The process in the latter graph has the advantage of being much easier to analyze; and, as we show, it gives an upper bound on the outcome of the original process.

Formally, let M_t be the number of infected nodes at time t in the original graph, where as before we count rounds from the start of Phase II. Let M'_t be the number of infected nodes at time t in an infinite graph under the assumptions that (a) no alert messages are ever sent out by the detector nodes, even though they are alerted by worm messages, and (b) each infected node spreads the worm to β unique, previously uninfected nodes in the network at each round. Where no confusion will result, we also use M_t and M'_t to refer to the set of nodes infected in each case.

Observe that the assumptions for M'_t only increase the number of infected nodes; so that M'_t *stochastically dominates* M_t in the sense that $\forall\ k \geq 0$, $Pr(M'_t \geq k) \geq Pr(M_t \geq k)$, no matter what strategy the worm applies in the original graph.

Let M_0 and M'_0 count the nodes infected by the end of Phase I, in their respective simulations. From Lemma 1, we have that $M_0 \leq |Z| - A_0 \leq \ln n$.

Lemma 2. *For all* $t \geq 0$*, the expected value of the random variable* M'_t *at time* t *is equal to* $(1 + \beta(1 - \gamma))^t M_0$.

Proof. By our assumption about the number of messages sent by the infected nodes and the fraction of detector nodes, the expected number of new infected nodes is $\beta(1-\gamma)\,\mathrm{E}[M'_t]$, where $(1-\gamma)$ is the probability that a given node is not a detector node. Hence the recurrence relation for $\mathrm{E}[M'_t]$ is $\mathrm{E}[M'_t]$=$(1+\beta(1-\gamma))\,\mathrm{E}[M'_{t-1}]$. Hence $\mathrm{E}[M'_t] = (1+\beta(1-\gamma))^t M_0$.■

We now show that M'_t remains closely bounded around its expected value, thus giving an upper bound on the variable M_t. The proof of the following lemma is somewhat technical; it is omitted from this extended abstract due to space constraints.

Lemma 3. *For any $c > 0$ and fixed β and γ, there exists a constant k such that, for sufficiently large n and any t, it holds that $M'_s \leq k\,\mathrm{E}[M'_s]$ for all $s \leq t$*

We now turn to alerted nodes. Let A_t be the number of nodes that are in the alerted state at time t. For any set of vertices A, let $N(A)$ be the set of neighbors of nodes in A in the overlay network that are not themselves in A. Let the random variable Z_t be equal to the number of nodes in $N(A_{t-1})$ that receive an alert message at time step t.

Lemma 4. *For all $t \geq 0$, $A_t \geq A_{t-1}$ + Z_t- M'_t*

Proof. Out of the unalerted nodes which receive alert messages, at most M'_{t-1} nodes could be infected nodes. Hence the lower bound result holds true. ■

Lemma 5. *For all $t \geq 0$, $E(Z_t) \geq (c\alpha/d)A_{t-1}$.*

Proof. Let S_{t-1} be the set of nodes that are alerted at time $t-1$ and let $n' = |N(S_{t-1})|$. Number the nodes in $N(S_{t-1})$ from 1 to n'. Let $X_{i,t} = 1$ if the i-th such node is alerted at time step t for the first time, and 0 otherwise. Then Z_t $\geq \sum_{i=1}^{n'} T(i,t)$. By linearity of expectation, $\mathrm{E}[Z_t] \geq \sum_{i=1}^{n'} \mathrm{E}[X_{i,t}]$. Observe that each node counted in A_{t-1} sends an alert to fixed neighbor with probability α/d; it follows that for each node i in $N(S_{t-1})$, $\Pr[X_{i,t} = 1] \geq \alpha/d$. We thus have $\mathrm{E}[Z_t] \geq n'\alpha/d \geq (c\alpha/d)A_{t-1}$, where c is the expansion factor. ■

Lemma 6. *For all $t \geq 0$ $A_t \geq A_{t-1} + (1/2)E(Z_t) - M'_t$.*

Proof. We now imagine that the alerted nodes use the following process to decide where to send out their α alert messages. They randomly permute all of their neighbors and then send out alerts to the first alpha nodes in this random permutation. Imagine further that some alerted node j determines its random permutation by assigning a random variable $X_{j,i}$ to each node i that is a neighbor of j. This random variable takes on a value uniformly at random in the real interval between 0 and 1. The nodes that the alert is sent to are thus determined by finding the α random variables among the d whose outcomes are closest to 0. For each node i and j, there is a separate such random $X_{j,i}$ and we note that these random variables are all independent. Let f be a function such that $Z_t = f(X_{1,1}, X_{1,2}, \ldots, X_{m,d})$. We note that f satisfies the Lipchitz condition, i.e

$|f(X_{1,1}, X_{1,2}, \ldots, X_{l,p}, \ldots, X_{m,d}) - f(X_{1,1}, X_{1,2}, \ldots, X'_{l,p}, \ldots, X_{m,d})| \le 1$. This is the case since a change in the outcome of a single $X_{i,j}$ will at most cause one new node to receive an alert and one old node to not receive an alert. Hence we can use Azuma's Inequality to say that Pr($Pr(|Z_t - E(Z_t)| \ge (1/2)E(Z_t) \le 2e^{-\frac{(1/4)E(Z_t)^2}{2A_{t-1}d}}$. Since by the previous lemma $E(Z_t) \ge (c\alpha/d)A_{t-1}$, the right hand side is less than or equal to $2e^{-\frac{((c\alpha/d)A_{t-1})^2}{8A_{t-1}d}}$ which is $O(1/n^{k'})$ for some constant $k' > 0$ since A_{t-1} is $\theta(\ln n)$. The lemma then follows by a simple Union bound. ■

Let k be the multiplicative constant of the expectation, in the statement of lemma 3.

Lemma 7. *For all $t \ge 0$, $A_t \ge (1 + (\alpha c)/(2d))A_{t-1} - k(1 + \beta(1-\gamma))^t \ln n$*

Proof. From Lemma 5 and Lemma 6 we get that the number of nodes alerted at round t follows the inequality $A_t \ge A_{t-1} + (1/2)((c\alpha/d)A_{t-1}) - M'_t$. Hence $A_t \ge (1 + (\alpha c)/(2d))A_{t-1} - M'_t$. By Lemma 2 and Lemma 3 we know that M'_t is no more than k$(1 + \beta(1-\gamma))^t \ln n$ for t rounds, with probability at least 1-t/n^c. Hence replacing the upper bound value of M_t in the above expression yields the inequality $A_t \ge (1 + (\alpha c)/(2d))A_{t-1} - k(1 + \beta(1-\gamma))^t \ln n$. ■

Let $p = (1 + (\alpha c)/(2d))$, $q = (1 + \beta(1-\gamma))$. Hence the recurrence relation as given in the last lemma is $A_t \ge pA_{t-1} - kq^t$.

Lemma 8. *For all $t \ge 0$, $A_t \ge p^t A_0 - k(q^t + pq^{t-1} + \ldots p^t)$*

Proof. Proof is by induction on t. It is easy to see that the base case holds. Assume that the claim holds for all rounds less than or equal to t-1. Hence $A_t \ge p(p^{t-1}A_0 - k(q^{t-1} + \ldots p^{t-1})) - kq^t$. Expanding the algebraic expression, we get the expression in the claim. ■

Let $\kappa = p/q$. Then $A_t \ge p^t \ln n - p^t k(1 + 1/\kappa + \ldots (1/\kappa)^t)$. Or

$$A_t \ge p^t(\ln n - k(1 + 1/\kappa + \ldots (1/\kappa)^t)). \tag{1}$$

2.3 Analysis of Phase III

In this phase, we make use of a graph with two types of expansion. We show below that a random d regular graph has the types of expansion that we need. The proof of the following two theorems are omitted from this extended abstract.

Theorem 1. *Let $d \ge 30$ and $\epsilon > 0$, then with high probability, a random d-regular graph G has the following properties*

1. *For any set S such that $\epsilon \log n \le |S| \le \frac{n}{d^2}$, $|N(S)| \ge |S|(\frac{d}{5} - 1)$.*
2. *For any set S such that $\frac{n}{d^2} \le |S| \le \frac{n}{2}$, $|N(S)| \ge \frac{|S|}{2}$.*

The following theorem assumes that the overlay network has expansion properties as given in the Theorem 1.

Theorem 2. *Assume that at some point, the number of alerted nodes is at least n/d^2 and that the number of infected nodes is no more than $n^{1-\epsilon}$ for some $\epsilon > 0$. Then w.h.p, at the end of the process, all but $o(n)$ nodes will be alerted.*

The next theorem is the main result of this section.

Theorem 3. *If $d \geq \alpha$ and $\frac{\alpha}{\beta(1-\gamma)} > \frac{2d}{c}$, then the algorithm RANDOM ensures that, w.h.p, only $o(n)$ nodes are ever infected.*

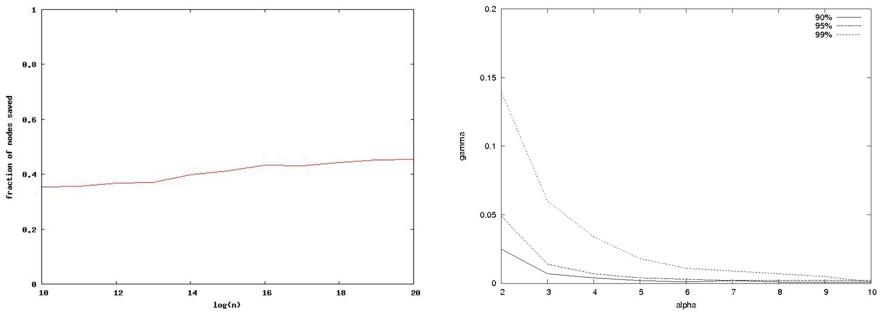
Proof. Since $\frac{\alpha}{\beta(1-\gamma)} > \frac{2d}{c}$, therefore $\frac{\alpha c}{2d} > \beta(1-\gamma)$. Hence $1+\frac{\alpha c}{2d} > 1+\beta(1-\gamma)$, or $p/q > 1$. From equation 1 it is clear that $A_t \geq p^t \ln n - 3k$. Hence $A_t \geq p^t$. Hence for $t \geq log_p n$, $A_t \geq \Omega(n)$. Hence in Phase II, the process cannot last for more that $log_p(n)$ steps. Hence from Lemma 3, we know that $M_{log_p(n)} \leq k(1+\beta(1-\gamma))^{log_p(n)}$ with probability greater than $1 - log_p(n)/n^c$. Hence $M_{log_p n}$ $<$ k $q^{log_p(n)}$. Since $p > q$, clearly $M_t = o(n)$ at the end of Phase II. Further it is O($n^{1-\epsilon}$). Now, from Theorem 2 , we know that if we have o($n^{1-\epsilon}$) infected nodes at the end of Phase II , we would have at most o(n) infected nodes at the end of the Phase III. ■

3 Empirical Results

We simulated the spread of a worm and an alert through a network to empirically determine the fraction of nodes saved.[5] We performed our experiment using a random d-regular graph as the overlay network and set each node in the network to be a detector node independently with probability γ. In addition, we fixed the worm strategy such that each infected node, in each round, sent out the worm to β unique nodes selected uniformly at random, and we fixed the alert strategy such that each alerted node sent out the alert to α unique nodes selected uniformly at random among its neighbors in the overlay network. We note that the worm strategy we used in these experiments is not necessarily the best possible worm strategy, but we selected this strategy for concreteness. Our d regular random graph was created using the configuration model method proposed in [2].

In each round we iterate through the set of vertices, allowing each infected or alerted node to send the worm or alert to the appropriate number of other nodes in the network. There are several possible strategies for resolving the status of a virgin (i.e. neither alerted or infected) node that gets both a worm message and an alert message in the same round. In our previous theoretical analysis, we assumed that if a node receives just one worm message it becomes infected. However, in our experiments, we used the somewhat more relaxed and realistic assumption that the probability that the node gets infected equals the number

[5] All of the code necessary to replicate these experiments is available at `http://www.cs.unm.edu/~navin/worm.html`.



(a) **(b)**

Fig. 1. (a) log of the network size versus fraction of nodes saved (b) contour plot of α versus γ required to save 99%, 95% and 90% of the nodes.

of worm messages received divided by the total number of messages received, and that the probability the node becomes alerted is 1 minus this quantity. We note that this assumption is equivalent to assuming that the messages all arrive in the node's message queue according to some random permutation.

Figure 1(a) illustrates our results when $\gamma = 0.1$, $\beta = 1$, $\alpha = 1$ and $d = 10$, where we varied the value of n from 2^{10} to 2^{20}, multiplying at each step by 2. To remove noise in the simulation, each data point represents the average over 100 trials. The best result we obtained was saving only 45% of the nodes for $n = 2^{20}$. Even though this final data point is somewhat disappointing, we do observe a clear increasing trend in the fraction saved as n increases.

Given these results, it seems for current network sizes, there is not much hope for the alert when $\alpha = \beta$. We thus next considered the case where $\alpha > \beta$. In practice, this condition may hold since the alerts are traveling through a predetermined overlay network and a technique such as throttling can ensure that alert messages received through the overlay are given priority over types of messages. To explore this scenario, we conducted experiments where we fixed β at 1. We then determined necessary values of γ for each α ranging from 2 to 10, that would ensure that we save 90%, 95% and 99% of the nodes (Figure 1(b)). The values of n and d used in the experiment were 10^6 and 100 respectively. The results of these experiments were much more encouraging. In particular, for $\alpha = 2$, we were able to save 99% of the nodes with $\gamma = .14$. When $\alpha = 5$, we required a γ of .018 to save 99% of the nodes, and when $\alpha = 10$, we required a γ of only .001 to save 99% of the nodes. These results suggest that our algorithms for spreading alerts might be most effective in conjunction with other techniques (like throttling) that would enable the alerts to spread more quickly than the worm.

4 Is expansion necessary?

In this section, we consider what happens in graphs with poor expansion properties. In particular, we look at the growth rate of the number of nodes at distance k from some initial point of infection, and show that if this growth rate is small, the worm successfully infects almost every node that does not detect it itself.

For the purposes of this lower bound, we adopt a simplified deterministic version of the model. We proceed in a sequence of rounds starting from the time at which the worm is first detected, and think of the graph as organized in layers V_0, V_1, $\ldots$, where V_0 contains the initial a_0 alerted and b_0 infected nodes, and each V_i is the set of nodes at distance i from this initial set.

We ignore the structure of the interconnections between layers; instead, we allow an SCA that has already alerted a_i nodes in layer V_i to alert any αa_i nodes in layer V_{i+1} in one round. Because the worm can spread without regard to the layer structure, we assume that it can attempt to infect these nodes first; a round thus consists of the worm attempting to infect nodes in layer V_{i+1} followed by the SCA attempting to alert any nodes that are left.

Let b_i be the total number of infected nodes in layer i after round i and let $B_i = \sum_{j=0}^{i}$ be the total number of infected nodes after round i without regard to what layer they are in. The worm can attempt to infect up to βB_i nodes in round $i+1$; of these, $\gamma \beta B_i$ will trigger detectors.

If we similarly let a_i be the number of alerted nodes in layer V_i after round i, then the SCA can attempt to alert αa_i nodes in layer V_{i+1}. But because the worm goes first, there may not be any nodes left to alert.

The overall pattern in round $i+1$ is thus:

1. The worm attempts to infect up to βB_i nodes in layer V_{i+1}, of which $(1-\gamma)\beta B_i$ become infected and $\gamma \beta B_i$ become alerted.
2. The SCA spreads from layer V_i to layer V_{i+1}, yielding an additional $\min(\alpha a_i, |V_{i+1}| - \beta B_i)$ alerted nodes.

This gives us the recurrence

$$\begin{aligned} b_{i+1} &= (1-\gamma) \min\left(|V_{i+1}|, \beta B_i\right) \\ a_{i+1} &= \gamma \min\left(|V_{i+1}|, \beta B_i\right) + \min\left(\alpha a_i, |V_{i+1}| - \beta B_i\right) \end{aligned}$$

Theorem 4. *Define a_i, b_i, and V_i as above. Let $|V_0|, |V_1|, \ldots$ be such that, for all $i \geq 0$,*

$$|V_{i+1}| \leq \beta(1-\gamma) \sum_{j=0}^{i} |V_i|.$$

Let $b_0 \geq (1-\gamma)|V_0|$. Then $b_i \geq (1-\gamma)|V_i|$ for all i.

Proof. Straightforward induction on i. The base case is given. For the induction step suppose the claim holds for i. Then we have

$$\begin{aligned}
b_{i+1} &= (1-\gamma)\min\left(|V_{i+1}|, \beta B_i\right) \\
&= (1-\gamma)\min\left(|V_{i+1}|, \beta \sum_{j=0}^{i} b_j\right) \\
&\geq (1-\gamma)\min\left(|V_{i+1}|, \beta(1-\gamma)\sum_{j=0}^{i} |V_j|\right) \\
&= (1-\gamma)|V_{i+1}|.
\end{aligned}$$

In other words, if the growth rate of the graph is small enough and the initial set of alerted nodes is small enough, then the SCA has no effect beyond the original detection sites.

For a large enough graph, a higher initial growth rate or lower initial worm numbers can be compensated for in the limit. For simplicity, we consider an *infinitely large* graph that is again organized into layers $V_0, V_1, \ldots$ as above.

Theorem 5. *Let a_i, b_i, V_i be as in Theorem 4. Let $b_0 > 0$ and let*

$$\limsup_{i\to\infty} \frac{|V_{i+1}|}{\sum_{j=0}^{i} |V_i|} < (1-\beta)\gamma. \tag{2}$$

Suppose further that $|V_{i+1}| \geq |V_i|$ for all i. Then

$$\lim_{i\to\infty} \frac{b_i}{|V_i|} = (1-\gamma).$$

Proof. We assume that α is sufficiently large that at the end of round i, any node in layer i that is not infected is alerted. This assumption only hurts the worm, so if the assumption is violated the result only improves.

From (2), there exists some ϵ, i_0 such that for all $i > i_0$, $|V_{i+1}| \leq (1-\epsilon)(1-\gamma)\beta\sum_{j=0}^{i}|V_j|$. Let $r_i = B_i/\sum_{j=0}^{i}|V_j|$ and compute, for $i > i_0$,

$$
\begin{aligned}
b_{i+1} &= (1-\gamma)\min\left(|V_{i+1}|, \beta B_i\right) \\
&= (1-\gamma)\min\left(|V_{i+1}|, \beta r_i \sum_{j=0}^{i} |V_i|\right) \\
&= \min\left((1-\gamma)|V_{i+1}|, r_i \beta (1-\gamma) \sum_{j=0}^{i} |V_i|\right) \\
&\geq \min\left((1-\gamma)|V_{i+1}|, \frac{r_i}{1-\epsilon}|V_{i+1}|\right) \\
&= \min\left(1-\gamma, \frac{r_i}{1-\epsilon}\right)|V_{i+1}|.
\end{aligned}
$$

Unless $r_i = 1-\gamma$, we expect $b_{i+1}/|V_{i+1}|$ to be larger than r_i; in particular we have $b_{i+1}/|V_{i+1}| \geq \min((1-\gamma),(1+\epsilon)r_i)$. The new ratio r_{i+1} is a weighted average of r_i and b_{i+1}/V_{i+1}. Under the assumption that $|V_i|$ is nondecreasing, the weight on the second term is at least $1/(i+1)$. Thus we have

$$
r_{i+1} \geq \frac{i}{i+1} r_i + \frac{\min(1-\gamma, \epsilon r_i)}{i+1} = \qquad r_i + \frac{\min((1-\gamma)-r_i, \epsilon r_i)}{i+1}.
$$

Observe that the first term in the minimum is decreasing and the second increasing. As long as $\epsilon r_i < (1-\gamma) r_i$, we have $r_{i+1} \geq r_i \frac{\epsilon}{i+1}$. So $r_{i+k} \geq r_i \left(1 + \epsilon \sum_{j=i}^{k-1} \frac{1}{j+1}\right)$; as the series diverges, eventually r_{i+k} must be large enough that the first term takes over. But then let $s_i = (1-\gamma) - r_i$, and compute $s_{i+1} = (1-\gamma) - r_{i+1} \leq s_i - \frac{s_i}{i+1} = s_i \frac{i}{i+1}$, from which it follows via a telescoping product that $s_{i+k} \leq s_i \frac{i}{i+k}$, which goes to zero in the limit. ■

The proof of the following theorem follows directly from the above.

Theorem 6. *For a graph with bounded degree d, we have $|V_{i+1}| \leq d \sum_{j=1}^{i} |V_j| + 1$. So if $(1-\gamma)\beta > d$ we expect almost no non-detector nodes to be alerted.*

5 Conclusion and Future Work

We have described a simple distributed algorithm for spreading alert messages through a network during a worm attack and have proven that this algorithm protects all but a vanishingly small fraction of the network provided that the alerts spread through an overlay network with sufficiently good node expansion. Our algorithm is provably good no matter what strategy the worm uses to spread through the network. We have demonstrated empirically that this algorithm works effectively against a randomly spreading worm under conditions that may be reasonable for modern computer networks. Finally, we have shown that if the overlay network has poor expansion, then the worm will likely infect almost all of the non-detector nodes in the network. Many open problems remain including: (1) tightening the upper and lower-bounds for the expansion needed in the overlay network to save almost all of the nodes; (2) developing other models for the spread of a dynamic process and its inhibitor over a network, and finding provably good strategies in these models; and (3) further empirical study to determine the efficacy of deploying our algorithm in a real network.

References

1. Stephen Baker and Brian Grow. Gambling Sites, This Is A Holdup, 2005. http://www.businessweek.com/magazine/content/04_32/b3895106_mz063.htm.
2. Bela Bollobas. *Random Graphs*. Academic Press, 1985.
3. David Brumley, James Newsome, Dawn Song, Hao Wang, and Somesh Jha. Towards automatic generation of vulnerability-based signatures. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 2–16, 2006.
4. Colin Cooper, Martin Dyer, and Catherine Greenhill. Sampling regular graphs and a peer-to-peer network. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete algorithms (SODA)*, 2005.
5. Manuel Costa, Jon Crowcroft, Miguel Castro, Anthony Rowstron, Lidong Zhou, Lintao Zhang, and Paul Barham. Vigilante: End-to-end containment of internet worms. In *Symposium on Operating System Principles (SOSP)*, 2005.
6. Manuel Costa, Jon Crowcroft, Miguel Castro, and Antony Rowstron. Can we contain internet worms? In *Proceedings of the 3rd Workshop on Hot Topics in Networks (HotNets-III)*, 2004.
7. Aaron Davis. Computer Worm Snarls Web, 2004. www.bayarea.com/mld/mercurynews/5034748.html.
8. Martin Garvey. Phishing Attacks Show Sixfold Increase This Year, June 2005. http://www.informationweek.com/story/showArticle.jhtml?articleID=164302582.
9. Ashlesha Joshi, Samuel King, George Dunlap, and Peter Chen. Detecting past and present intrusions through vulnerability-specific predicates. In *Symposium on Operating System Principles (SOSP)*, 2005.
10. Robert O'Harrow Jr. Internet Worm Unearths New Holes, 2003. www.securityfocus.com/news/2186.
11. Robert Lemos. Slammer Attacks May Become Way of Life for the Net, 2003. http://www.news.com/2009-1001-983540.html?tag=fd_lede2_hed.
12. John Leyden. Phishers Tapping Botnets to Automate Attack, 2004. http://www.theregister.co.uk/2004/11/26/anti-phishing_report/.
13. John Leyden. ISPs urged to throttle spam zombies, 2005. http://www.theregister.co.uk/2005/05/24/operation_spam_zombie/.
14. Zhenkai Liang and R. Sekar. Fast and automated generation of attack signatures: a basis for building self-protecting servers. In *Proceedings of the 12th ACM Conference on Computer and Communications Security (CCS)*, pages 213–222, 2005.
15. Dan Liet. Most Spam Generated by Botnets, Says Expert, 2004. http://news.zdnet.co.uk/internet/security/0,39020375,39167561,00.htm.
16. L.Zhou, L.Zhang, F.McSherry, N.Immorlica, M.Costa, and S. Chien. A first look at peer-to-peer worms: Threats and defenses. In *International Symposium on Peer-to-peer Systems (IPTPS)*, 2005.

17. D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. Inside the Slammer Worm. *IEEE Security and Privacy journal*, 1(4):33–39, 2003.
18. Roberto Preatoni. Prophet Mohammed protest spreads on the digital ground. Hundreds of cyber attacks against Danish and western webservers spreading rage in the name of Allah, February 2006. http://213.219.122.11/en/news/read/id=205987/.
19. Paul Roberts. Al-Jazeera hobbled by DDOS attack, 2003. http://www.infoworld.com/article/03/03/26/HNjazeera_1.html.
20. Antony I. T. Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg*, pages 329 – 350, 2001.
21. S. Shakkottai and R. Srikant. Peer to peer networks for defense against internet worms. In *Proceedings of the 2006 workshop on Interdisciplinary systems approach in performance evaluation and design of computer and communications sytems*, 2006.
22. Eugene Spafford. Exploring Grand Challenges in Trustworthy Computing. http://digitalenterprise.org/seminar/spafford2.html.
23. Will Sturgeon. Denial-of-service-attack victim speaks out, 2005. http://www.zdnetasia.com/insight/business/0,39044868,39233051,00.htm.
24. Chris Talbot. Phishing Attacks Up More Than 200% in May, says IBM, 2005. http://www.integratedmar.com/ecl-usa/story.cfm?item=19703.
25. M. Vojnovic and A. Ganesh. On the effectiveness of automatic patching. In *ACM Workshop on Rapid Malcode (WORM)*, 2005.