Analysis of the Worm Detection problem

James Aspnes * Jared Saia † Vishal Sanwalani ‡ Navin Rustagi §

*Department of Computer Science, Yale University email: aspnes@cs.yale.edu
†Department of Computer Science, University of New Mexico, Albuquerque, NM 87131-1386; email: saia@cs.unm.edu.
‡Microsoft research; email: vsanwalani@gmail.com.
§University of New Mexico;email:navin@cs.unm.edu

1 Introduction

Modern computer worms propagate too quickly for human detection. The recent slammer worm infected 90% of vulnerable hosts within 10 minutes[3]. Since attacks are now occuring 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 cert ifying alerts(SCAs)[1]. 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 automati cally 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 [2] 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. The Vigilante system uses the Pastry peer to peer overlay network to efficiently distribute the SCAs. It is shown empirically that a small fraction of detectors(. 001) is enough to ensure that a worm infects no more than 5% of the vulnerable population

While these initial results are promising, several open problems remain in the area of SCA deployment. First, Vigilante requires a trusted onine certification authority to participate in the network in order to prevent sybil attacks. Second, Vigilante requires that the overlay network consist only of superpeers. Superpeers are dedicated machines which run only the overlay code and the SCA verifying code. Each peer in the network connects to two superpeers in the overlay, but to no other peers. This ensures that the worm can not learn and exploit the topology of the overlay network. These two assumptions seem to be valid only for a SCA distribution system that is owned and operated by a single company. A final problem with the Vigilante system is that while it performs well empirically, the system has no theoretical guarantees.

2 Model

We start with a synchronous complete network, in which any node can communicate with any other node in one time unit. A *worm* starts spreading through the network starting from an arbitrary initial node. At each step, each copy of the worm can attempt to infect a constant number of previously uninfected nodes. If the target node possesses a *detector*, it is not infected; instead, it generates a *self-certifying alert* (SCA) that protects any node that receives it from copies of the original virus.

Self-certifying alerts also spread through the network: each node in possession of an SCA can choose to distributed it to all of its neighbors at each step. However, unlike the virus, which can jump to arbitrary nodes, SCAs can only travel over a pre-existing overlay network consisting of a subgraph of the original network.

An edge in this overlay network represents an agreement between two nodes to accept SCAs from each other; we assume that the SCA-spreading algorithm is "polite" in the sense that it does not bombard arbitrary nodes with SCAs 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.

2.1 Assumptions about the model:

- Assume that γ fraction of the total nodes are the detector nodes, i.e they cannot be infected by the worm message. The detector nodes are spread uniformly at random across the network. In other words if a message reaches a node then with probability γ it cannot be infected because it is a detector node. Moreover when a detector node recieves a worm message, it becomes alerted.

- The overlay network is a d regular graph. We have this assumption because w.h.p a random d regular graph is an expander. Hence we assume that the overlay network has expansion properties, with the expansion parameter as c.

- For each alerted node, it sends alerts to α nodes. Alerts are only sent to the neighbors in the overlay network.

- For each infected node, it sends the worm to β nodes. Therefore if there are m infected nodes then in the next time step, a total number of βm messages are sent out.

- Worm knows the network structure, overlay network structure, and SCA algorithm if any (though not SCAs coin flips). Worm doesnt know where the detectors are, but the worm can use the worst possible way to attack the network, without knowing the distribution of the detector nodes. For example it could always try to infect nodes which have never been infected before.

3 SCA vs random worm

We assume that a single alerted node sends out α messages in parallel and that the degree of each node in the overlay network is d. In this paper we show that w.h.p , we are able to save most of the nodes from getting infected. More precisely at the end of the process only o(n) nodes get infected, and all other nodes get alerted. To show this, we will divide the process of the spread of the worm and SCA into three phases. In phase I, we start with a single node that is infected. This phase ends when $\ln n$ nodes are infected. We give a headstart to the adversary of $\ln n$ nodes by assuming that none of the detector nodes have recieved the worms, hence no node has been alerted. Phase II starts at the end of phase 1 and it ends when there is a set of λn nodes that have received one of either the worm message or the SCA message, for some $0 < \lambda < 1/2$. Phase III starts at the end of phase 2 and ends when all nodes have received the worm or SCA message.

3.1 Analysis of phase II

Let M_t be the number of infected nodes at time t. By our assumption M_0=ln n. Let M'_t be the number of infected nodes at time t if no alert messages are ever sent. We say that M'_t *stochastically dominates* M_t if $\forall\ k \geq 0$, $Pr(M'_t \geq k) \geq Pr(M_t \geq k)$. In this case it is clear that indeed M'_t stochastically dominates M_t, just by the definition of M'_t and M_t.

LEMMA 3.1. *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 $\beta(1 - \gamma)E(M'_t)$, where $(1 - \gamma)$ is the probability that a given node is not a detector node. Hence the recurrence relation for $E(M'_t)$is $E(M'_t)$=$(1 + \beta(1 - \gamma))E(M'_{t-1})$. Hence $E(M'_t) = (1 + \beta(1 - \gamma))^t M_0$. ■

We now show that M'_t remains closely bounded around its expected value, thus giving an upperbound on the variable M_t.

LEMMA 3.2. *For all $t \leq x$, the random variable M_t is less than or equal to $(1 + \delta)E(M'_t)$ with probability greater than (1-x/n), where $0 \leq \delta \leq 1/2$ and $\delta^2\beta(1 - \gamma)(1 + \beta(1 - \gamma))^{t-1} \geq 3$*

Proof. To show that M_t is bounded above by $(1 + \delta)\ E(M'_t)$, it is sufficient to show that M'_t concentrated around $(1 + \delta)E(M'_t)$, because M'_t stochastically dominates M_t.

Let S_t be the number of nodes that recieve the worm message at timestep t, assumng that no alert messages are ever sent. Let $X(v, t) = 1$ if the node v becomes infected for the first time at time t, 0 otherwise. Therefore Pr(X(v, t)=1)=1-γ. Define $Y_t = M'_t - M'_{t-1}$, or the number of bad nodes that have been infected at time step t. Clearly $Y_t = \Sigma_{i \in S_t} X(i, t)$. Hence Y_t is a sum of indentically distributed Poisson Variables. From the previous section, we know that $E(M'_t) = (1 + \beta(1 - \gamma)^t)M_0$, hence E($Y_t$)=$(1 + \beta(1 - \gamma))^t M_0 - (1 + \beta(1 - \gamma))^{t-1} M_0$. Let the right hand side of the previous expression be written equivalently as $c_0 M_0$, or $c_0 \ln n$, where c_0 is a function in t. Therefore by *Chernoff bounds*,

$$Pr[Y_t \geq (1 + \delta)E(Y_t)] \leq e^{\frac{-\delta^2\mu}{3}}, 0 \leq \delta < 1$$

Hence Pr$[Y_t > (1 + \delta)E(Y_t)]$

$$\leq \frac{1}{e^{(\delta^2 c_0 ln(n))/3}} \tag{3.1}$$

$$= \frac{1}{n^{(\delta^2 c_0)/3}} \tag{3.2}$$

Hence whenever $\delta^2\beta(1 - \gamma)(1 + \beta(1 - \gamma))^{t-1} \geq 3$, then Pr$[Y_t > (1 + \delta)E(Y_t)] \leq 1/n$.

We have shown above that for each t, probability that M'_t varies from $E(M'_t)$ by a variance of δ is less than 1/n. Now we show that the probability that M'_t varies from the expected value by δ, even once is o(1) for sufficiently small values of x, so that even if the low probability event occurs, the probability that it would have any effect at the end of the process is o(1).

Fix a $\delta > 0$. Let ξ_i be the event that $|Y_i - E(Y_i)| \geq \delta E(Y_i)$. Let ξ be the event such that $\xi = \xi_1 \cup \ldots \xi_t$, i.e the event that at least one of the events ξ_i occurs. By the union bound $Pr(\xi) \leq \Sigma_{i=1}^{t} Pr(\xi_i)$. Since each of the $\xi'_i s$ has probability less than $\frac{1}{n}$, the probabilty that ξ occurs is x/n since the process lasts for x steps. Hence the probability of Y_t, varying from its expectation by a δ fraction is o(1), for small enough x.

■

Now we analyse the number of alerted nodes in this process. Let A_i be a random variable giving the number of alerted nodes at time i. We show that A_0=$\theta(\ln n)$

LEMMA 3.3. *At the start of Phase II, $\theta(\ln n)$ nodes are alerted with probability greater than $(1 - 1/n^c)$, $0 < c \leq 1$*

Proof. The number of worm messages sent out at the beginning of Phase II is $\beta \ln n$. The probability that any of these messages alerts a detector node is γ. Hence the expected number of nodes that are alerted at the beginning of Phase II is $\gamma\beta \ln n$.

Let $X_{v,i}$=1 if the node v is a bad node and sends the ith message to a detector node, 0 otherwise. Clearly the number of alerted nodes A_0 at the start of Phase II is a function F of $X'_{v,i}s$, where v is a bad node and $0 < i < \beta$, s. t $|F(X_{v_1,1}, X_{v_1,2}, \ldots, X_{v',t'}, \ldots X_{v_{\ln n},\beta}) - F(X_{v_1,1}, X_{v_1,2}, \ldots, X_{v",t"}, \ldots X_{v_{\ln n},\beta})| \leq 1$. Hence by a version of Azuma's inequality $Pr(Pr|A_0 - E(A_0)| \geq 1/2E(A_0)) \leq 2e^{-\frac{E(A_0)^2/4}{2 \ln n\beta}}$=$2e^{-O(\ln n)} \leq 1/n^c$, $0 < c \leq 1$. Hence with high probability the number of nodes alerted will be $\theta(\ln n)$. ■

Let N(A) be the set of nieghbours of A, not in A on the underlying network graph, for a subset of vertices A. Let the random variable Y_t be equal to the number of nodes in $N(A_{t-1})$ that recieve an alert message at time step t. Let ΔA_t=$Y_t - M'_{t-1}$.

LEMMA 3.4. *For all $t \geq 0$, $A_t \geq A_{t-1} + \Delta A_t$*

Proof. Out of the nodes which recieve alert messages, M'_{t-1} nodes could be bad nodes. Hence the lower bound result holds true ■

LEMMA 3.5. *For all $t \geq 0$, $E(Y_t) \geq \alpha/d(c(A_{t-1}) - M'_{t-1})$*

Proof. Let $T_{i,t} = 1$ if the ith node is alerted at time step t for the first time, 0 otherwise Clearly $Y_t \geq \Sigma_{i=0}^{i=n} T(i,t)$. Therefore by linearity of expectation, $E(Y_t) \geq \Sigma_{i=0}^{i=n} E(T_{i,t})$. $Pr(T_{i,t} = 1) \geq \alpha/d$, because the probability that a node is among the d nieghbors of an alerted node that get at least one of the α messages sent out is α/d. The total number of nodes which are alerted in time step t are atleast $c(A_{t-1}) - M'_{t-1}$. Hence the result. ■

LEMMA 3.6. *For all $t \leq x$, $A_t \geq A_{t-1} + (1 - \delta)E(Y_t)$ - M'_{t-1}, for $0 \leq \delta \leq 1/2$, with probability greater than or equal to $1 - x/n$.*

Proof. Let A_{t-1}=m. Let $X_{j,i}$= 1 if node j in A_{t-1} sends an alert to its ith nieghbour, 0 otherwise. Let F be a function such that Y_t=$F(X_{1,1}, X_{1,2}, \ldots, X_{m,d})$. Clearly F is a function which follows the Lipchitz condition, i.e $|F(X_{1,1}, X_{1,2}, \ldots, X_{l,p}, \ldots, X_{m,d}) - F(X_{1,1}, X_{1,2}, \ldots, X'_{l,p}, \ldots, X_{m,d})| \leq 1$. Note that $X_{j,i}$ variables are independent of each other. Therefore by a version of Azuma's Inequality Pr($Pr(|Y_t - E(Y_t)| \geq 1/2E(Y_t) \leq 2e^{-\frac{1/4E(Y_t)^2}{2md}}$. Since by the previous lemma $E(Y_t) \geq kA_{t-1} + f$, k a constant and f an upperbound for M'_{t-1}. Hence the right hand side is less than or equal to $2e^{-\frac{1/4(k^2A_{t-1}^2 + f^2 + 2kfA_{t-1})}{2md}} \leq \theta(1/e^m)$, as A_{t-1} takes on the value m. Since $m \geq \ln n$, right hand side is less than 1/n.

Fix $\delta > 0$ Let ξ_i be the event that $|Y_i - E(Y_i)| \geq \delta E(Y_i)$. Let ξ be the event such that $\xi = \xi_1 \cup \xi_2 \cup \ldots \xi_t$, i.e the event that at least one of the events ξ_i occurs. By the Union bound $Pr(\xi) \leq \Sigma_{i=1}^{t} Pr(\xi_i)$. Since each of the $\xi'_i s$ has probability less than $\frac{1}{n}$, the probabilty that ξ occurs is x/n.

■

4 SCA's to the rescue

THEOREM 1. *If* $((\alpha c)/d - 2\beta(1-\gamma)) = 1$, $\alpha c - 2\alpha \geq 2d$, *and* $1 + \beta(1-\gamma) \leq$ *2, then the number of nodes infected at the end of the process, i.e when every node has either recieved a worm message or an alert message, is o(n) with probability greater than* $1 - log(n)/n$

Proof. Let us assume that the process runs for x steps. We also assign the worst possible value of δ for our analysis, to δ, i.e $\delta = 1/2$. From Lemma 3.5 and Lemma 3.6 we get that the number of nodes alerted at time step t follow the inequality $A_t \geq A_{t-1} + (1-\delta)(\alpha/d(c(A_{t-1}) - M'_{t-1})) - M'_{t-1}$.Hence $A_t \geq (1 + (\alpha c)/(2d))A_{t-1} - (1 + \alpha/(2d))M'_{t-1}$. By Lemma 3.1 and Lemma 3.2 we know that M'_{t-1} is closely bounded around $(1 + \beta(1-\gamma))^{t-1} \ln n$ with probability at least (1-x/n). Hence replacing the upper bound value of M_{t-1} in the above expression yields the inequality $A_t \geq (1 + (\alpha c)/(2d))A_{t-1} - (1 + \alpha/(2d))(1 + \beta(1-\gamma))^{t-1} \ln n$. Let $p = (1 + (\alpha c)/(2d))$, $q = (1 + \beta(1-\gamma))$ and let $k = (1 + \alpha/2d) \ln n$. Hence the recurrence relation is $A_t \geq pA_{t-1} - kq^{t-1}$.

LEMMA 4.1. *For all* $t \geq 0$, $A_t \geq p^t A_0 - k(q^{t-1} + pq^{t-2} + \ldots p^{t-1})$

Proof. Proof is by induction on t. It is easy to see that the base case holds. Assume that the claim holds for all time steps less that or equal to t-1. Hence $A_t \geq p(p^{t-1}A_0 - k(q^{t-2} + \ldots p^{t-2})) - kq^{t-1}$. Expanding the algebraic expression, we get the expression in the claim . Hence proved ■

Now since q=p/2 by theorem statement, hence $A_t \geq p^t \ln n - kp^{t-1}(1 + 1/2 + \ldots 1/2^{t-1}) \geq p^t \ln n - kp^{t-1}2 \geq p^{t-1}(p \ln n - 2k)$. Hence if $p \ln n \geq 2k$, then $A_t = \Omega(p^{t-1})$. Therefore $(1 + (\alpha c)/(2d)) \ln n \geq 2((1 + \alpha/2d) \ln n)$ or $\alpha c - 2\alpha \geq 2d$, which is true by theorem statement. Let $x = log(n)$, therefore $A_t \geq (1 + (\alpha c)/2d)^{log(n)}$, or $A_t = \Omega(n)$. Therefore the process cannot last more that $log(n)$ time steps. Hence $M_{log(n)} \leq (1 + \beta(1-\gamma))^{log(n)} \ln n$. Hence if $(1 + \beta(1-\gamma)) < 2$, then $M_{log(n)} \leq o(n)$. Therefore at the end of the process, o(n) nodes will be infected and $\Omega(n)$ nodes will be alerted. Hence proved. ■

References

[1] M.Costa, J.Crowcroft, M.Castro, and A.Rowstron. Can we contain internet worms? In Proceedings of the 3rd Workshop on Hot Topics in Networks (HotNets-III), 2004.

[2] M.Costa, J.Crowcroft, M.Castro, A.Rowstron, L.Zhou, L.Zhang, and P.Barham. Vigilante: End-to-end containment of internet worms. In Symposium on Operating System Principles (SOSP), 2005.

[3] D.Moore, V.Paxson, S.Savage, C.Shannon, S.Staniford, and N.Weaver. Inside the slammer worm. IEEE Security and Privacy journal, 2003.

[4] L.Zhou, L.Zhang, F.McSherry, N.Immorlica, M.Costa, and Chien. A first look at peer-to-peer worms: Threats and defenses. In International Symposium on Peer-to-peer systems, 2005.