Worm Evolution Tracking via Timing Analysis

Moheeb Abu Rajab Fabian Monrose Andreas Terzis
Computer Science Department
Johns Hopkins University
{moheeb,fabian,terzis}@cs.jhu.edu

ABSTRACT

We present a technique to infer a worm's infection sequence from traffic traces collected at a *network telescope*. We analyze the fidelity of the infection evolution as inferred by our technique, and explore its effectiveness under varying constraints including the scanning rate of the worm, the size of the vulnerable population, and the size of the telescope itself. Moreover, we provide guidance regarding the point at which our method's accuracy diminishes beyond practical value. As we show empirically, this point is reached well after a few hundred initial infected hosts (possibly including "patient zero") has been reliably identified with more than 80% accuracy. We generalize our mechanism by exploiting the change in the pattern of inter-arrival times exhibited during the early stages of such an outbreak to detect the presence and approximate size of the hit-list. Our mechanism is resilient to varying parameters like the worm scanning rate and the size of the vulnerable population, and can provide significant insights into the characteristics of the hit-list even under spreading dynamics that exceed that of currently known worms. Lastly, to illustrate the practicality of our solution, we apply our approach to real-world traces of the Witty worm and provide a refined estimate on the previously suspected hit-list size.

Categories and Subject Descriptors

D.4.6 [**Operating Systems**]: Security and Protection—*Invasive Software*

General Terms

Security, Measurement

Keywords

network security, Internet worms, methods of attribution

1. INTRODUCTION

Worm outbreaks are security events that occur with relatively low frequency, but when they do occur, they can have significant impact on daily network operations. This ever-present threat of severe network disruption has been the motivating factor behind most, if not all, research on practical strategies for worm detection and containment (see [11, 16, 18, 20, 21]). There is, however, one desirable aspect of research that falls under the general umbrella of worm mitigation that has received far less attention in the past, namely back-tracking the evolution of a worm outbreak. In fact, thus far there has been little progress in the design and analysis of effective strategies for discovering the sequence with which a worm infected its victims. Even for worms that exhibit uniform scanning behavior, uncovering this sequence is a daunting task, but one that provides invaluable information. For one, doing so has direct pragmatic implications as it allows network operators to pinpoint the initial set of infected machines, thereby gleaning potentially useful forensic evidence.

Unfortunately, to date there have been few proposals for retracing the steps of a worm infection. Xie *et al.* offered a randomized approach that traces the origin of a worm attack by performing a random walk over the hosts contact graph [23]. The graph is generated by collecting traffic traces containing the list of hosts that contacted other potential victims during the worm's propagation. While this approach can provide a wealth of information about the worm's evolution, most notably, the who-infected-whom tree and *patient zero* (i.e., the initial victim), it requires traffic traces on a global scale to reconstruct the evolution of a large scale event. A different approach was suggested more recently by Kumar *et. al.* [8] where the Witty worm [15] was reverse engineered to recover the random scanning algorithm and corresponding initial seeds. Given knowledge of the target selection algorithm, the sequence of scans could be re-enacted to provide a detailed view of the worm's evolution, and also provide insights into characteristics of the infected hosts. However, although the information required for this approach (i.e., the payload) can be recovered locally, the mechanism can not be easily generalized to other worms, since each instance will have to undergo the same, possibly arduous, task of reverse-engineering.

In this paper, we address these limitations by exploring a different, and we believe more general, approach whereby we can infer the infection evolution from the history of worm scans seen at a *network telescope* [9]. Intuitively, if probes from an infected source arrive at the monitor[1] before scans from a different infected host, we can infer that the first host was infected before the latter one. However, factors such as the randomness inherent in the scanning process, the telescope size relative to the vulnerable population size as well as the stage of the worm propagation compound the issue of reliably inferring the evolution. Moreover, as the time difference between contacts at the telescope decreases, it becomes increasingly difficult to detect the actual infection sequence. Nonetheless,

[1]We use the terms monitor and network telescope interchangeably.

WORM'05, November 11, 2005, Alexandria, Virginia, USA.

we show that while the approach suggested here is relatively simple, it does lead to tangible results.

Our contributions in this paper are twofold. First, we provide an analytical model that expresses the fidelity of the infection evolution generated by the monitor as a function of the size of the vulnerable population, the size of the monitor, and the worm scanning rate. We validate the accuracy of our model using simulations under varying conditions. We argue that such simulations are necessary, as they are the only prudent method to accurately obtain the exact worm evolution and compare it against the one generated by the monitor. The results of our analysis are encouraging and they show that our technique is highly accurate in determining the initial set of infected hosts—including possibly identifying patient zero. Second, we provide a similarly intuitive approach for detecting the presence and size of a hit list. We do so solely on information gained by exploiting the change in the pattern of inter-arrival times exhibited during the early stages of a worm outbreak. We apply our technique to real-world traces of the Witty Worm [22] and show that it can reliably detect the existence and the size of the hit-list.

The rest of the paper is organized as follows: We present an overview of our techniques and analyze the accuracy of the telescope's view of the infection in Section 2. In Section 3 we evaluate the accuracy of the infection sequence provided by our method to the actual sequence through simulations. Section 4 explains how the existence and size of a potential list can be detected. We present previous work in Section 5 and conclude in Section 6.

2. ANALYTICAL MODEL

Our methodology exploits two invariant properties of worm behavior, namely that (*i*) worms spread by actively scanning the IP space[2] at random looking for vulnerable hosts to infect (see [3, 4, 10] for examples of past scanning worms) and (*ii*) worm spreading follows the classical characteristics of pathogen spreading in a fixed population — the worm onset starts with a slow spreading rate followed by a dramatic rise in the rate of infection after enough hosts have been infected [6]. With these invariants in mind, we estimate the initial worm evolution sequence and identify patient zero (or any initial hit list) by observing the order of scans and the pattern of inter-arrival times between successive first scans arriving at a network telescope.

The telescope's ability to correctly reconstruct the evolution sequence, relies however on a number of key elements including the telescope's size, the scanning rate of the worm, and the size of the vulnerable population. Here, we quantify the accuracy between the sequence generated from the point of view of a network telescope to the actual infection sequence of a simulated worm, and evaluate the extent to which the telescope can accurately reproduce this sequence.

In order to do so, we first need to distinguish between two primary quantities, namely, the average time to infect a new host, which we denote as T_{in}, and T_d, the time taken by the telescope to detect (with a particular confidence level α) a newly infected host. In general, with few scanners during the initial stages of the infection, the time to infect a new host will be relatively large compared to the time it takes any infected host to send its first scan to the telescope. Therefore, with high likelihood, a newly infected host will send at least one scan to the telescope before the worm is able to infect additional vulnerable hosts. The general idea we explore here (as shown in Figure 1) is that by observing the first scan from each unique source at the telescope, we can infer a close estimate of the actual worm initial infection sequence. The accuracy of this estimate, however, deteriorates as the worm progresses and T_{in} approaches or becomes less than T_d. However, as we show later, significant degradation in accuracy occurs long after we infer the initial worm evolution sequence with high confidence.

[2]To be accurate, the class of *scanning* worms from the taxonomy proposed in [19] propagate by active probing. Our analysis is nonetheless relevant since the majority of worms observed in the wild so far are scanning worms

V	Total number of vulnerable hosts
n_i	Number of infected nodes at time step i
s	Average scan rate (scans/time step) per infected node
M	The size of the telescope address space
T_{in}	The average time elapsed before subsequent infections from the vulnerable population
P_j	The probability of infecting at least one new host at time j
R_T	The number of trials needed in order to contact the telescope (at confidence level α)
T_d	Time to detect (with α confidence) at least one scan from a worm instance
p_e	Probability that more than one host is detected within detection window T_d
$Y_{\mathcal{B}\rightarrow\mathcal{A}}$	Weighted similarity between sorted sets $\mathcal{B}$ and $\mathcal{A}$
$r_{(i,\mathcal{B})}$	The rank of element i in sorted set $\mathcal{B}$
m	The length of the range over which the similarity between sets $\mathcal{B}$ and $\mathcal{A}$ is computed

Table 1: Notation.

Determining the presence (and size) of a hit list is accomplished by a similar straightforward conjecture — i.e., the inter-arrival pattern of hosts from this list contacting the telescope should exhibit different pattern compared to the inter-arrivals of hosts infected as the worm propagates. As we show in Section 4, when a hit list is used, the change in the inter-arrival pattern of new infected hosts exhibited near the onset of the worm spreading allows us to pinpoint the size of that list.

In what follows, we quantify both T_{in} and T_d then compute the likelihood of reconstructing the correct infection sequence at the telescope. To simplify the analysis we assume a constant scanning rate among all infected hosts. This assumption may be reasonable for TCP-based worms [12, 17, 24] where the scanning rate is limited by network delay rather than bandwidth. On the other hand, the scanning rate of UDP-based worms depends heavily on the link bandwidth and therefore exhibits significantly more inhomogeneity as shown by Witty [15]. We explore the effect of this inhomogeneity in Section 3.

The notation we use in the remainder of the paper is summarized in Table 1.

2.1 Infection and Detection Times

Consider a uniform scanning worm spreading with a per host scanning rate s over a vulnerable population of size V. We use a discrete time model in our analysis, following the model presented in [5]. In this model, the number infected hosts n_i, at the end of the i-th time step is given by:

$$n_i = n_{i-1} + (V - n_{i-1}) \left[1 - \left(1 - \frac{1}{2^{32}} \right)^R \right] \qquad (1)$$

where R is the total number of scans sent by the n_{i-1} infected hosts. The second term in Eq.(1) represents the increase in the

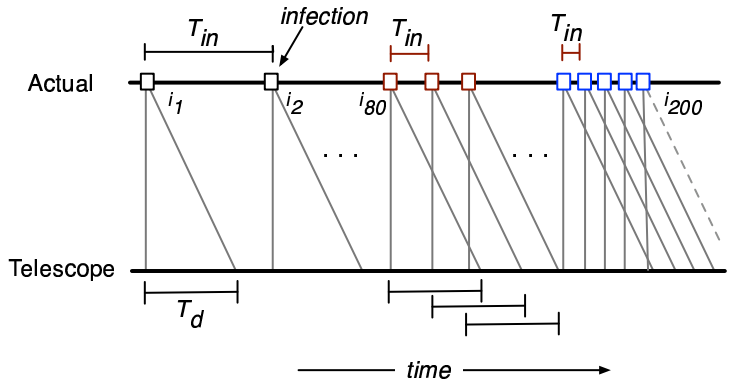

Figure 1: The view of the infection series order as seen by the telescope compared to the actual infection. When T_{in} is greater than T_d the telescope's view of the infection series will be very similar to the actual series. However, as T_{in} approaches T_d (i.e., as the worm spreads), the probability of mis-classifying the series order grows accordingly.

number of infected hosts. To compute the time T_{in} necessary to infect one additional host, we set this term to one:

$$(V - n_{i-1})\left[1 - \left(1 - \frac{1}{2^{32}}\right)^{T_{in} s n_{i-1}}\right] = 1 \tag{2}$$

Solving Eq.(2) for T_{in}, we have:

$$T_{in} = \frac{\log\left(1 - \frac{1}{V - n_{i-1}}\right)}{s n_{i-1} \log\left(1 - \frac{1}{2^{32}}\right)} \tag{3}$$

In computing the detection time by the telescope, rather than considering the average case, we are concerned with the point at which we can judge with confidence α that a newly infected host has indeed been detected by the telescope. The probability α that at least one scan from the host will reach the telescope from R_T scans is:

$$\alpha = 1 - \left(1 - \frac{M}{2^{32}}\right)^{R_T} \tag{4}$$

Solving Eq.(4) for R_T we get:

$$R_T = \frac{\log(1 - \alpha)}{\log\left(1 - \frac{M}{2^{32}}\right)} \tag{5}$$

and therefore, the time $T_d = R_T/s$, where s is the average scanning rate.

2.2 Telescope Accuracy

The relation between T_d and T_{in} defines the telescope's ability to reconstruct the actual sequence of the worm evolution. When $T_{in} > T_d$, the latest infected host will be detected with high likelihood by the telescope before any infected node is able to infect additional hosts from the vulnerable population. Therefore, the order of unique infected sources detected by the telescope will be close to the actual worm infection sequence. On the other hand, when T_{in} approaches or becomes lower than T_d, it is likely that more hosts will be infected within T_d, and these hosts may be detected out of order, thereby causing lower correspondence between the infection sequence observed at the telescope and the actual worm evolution.

To provide a measure of this degradation in accuracy between the two views, we compute the probability P_e that more that one infected hosts are detected by the telescope within the detection time T_d.

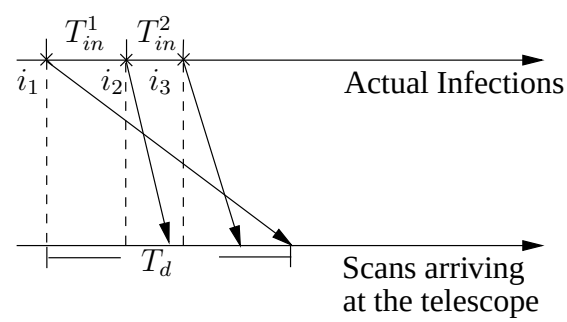

Figure 2: When $T_{in} < T_d$ it is possible for the telescope to see a scan from hosts i_2 and i_3 that were infected after i_1 before it sees a scan from i_1.

Figure 2 provides a visual example of this scenario. Considering the previous discussion, it is clear that if $T_d < T_{in}$, then $P_e = 0$ since i_1 will be detected by the telescope before the next host is infected. On the other hand, if $T_d > T_{in}$ it is possible that a host infected after i_1 will scan the telescope first. From Figure 2, we can see that the probability that none of the $(T_d - T_{in}^1) \cdot s$ scans from i_2 arrive at the telescope is:

$$\text{Prob[} i_2 \text{ is undetected]} = \left(1 - \frac{M}{2^{32}}\right)^{(T_d - T_{in}^1)\cdot s} \tag{6}$$

Similarly, the probability that i_3 will not be detected is:

$$\text{Prob[} i_3 \text{ is undetected]} = \left(1 - \frac{M}{2^{32}}\right)^{(T_d - T_{in}^1 - T_{in}^2)\cdot s} \tag{7}$$

P_e is then the probability that at least one of the n hosts infected within time T_d from the first host in the sequence (i_1 in the figure), sends at least one scan to the telescope, therefore:

Number of Vulnerable hosts	12,000
Average scanning rate per infected host (s)	350 scans/tick
Size of initial Hit List	1
Scanning Algorithm	Uniform
The telescope detection confidence α	95%
Network Delay (Normally distributed)	$\mu = 50$ ms $\sigma = 20$ ms

Table 2: Simulation Parameters .

$$P_e = 1 - \prod_{i=1}^{n} \left(1 - \frac{M}{2^{32}}\right)^{(T_d - \sum_{j=1}^{i} T_{in}^{j})s} \tag{8}$$

3. EVALUATION

In the remainder of the paper we validate the analytical model presented in Section 2 via simulation. The simulator we built allows us to simulate worms spreading over populations of vulnerable hosts of different size and density. It is also possible to simulate worms with different scanning rates and target selection strategies.

In the following experiments, unless otherwise specified, we use the simulation parameters from Table 2. We later vary these parameters to show their impact on the accuracy of the monitor's view.

3.1 Time to Infect and Time to Detect

To validate the correctness of our approach in inferring the worm evolution sequence, we first present the inter-arrival time of newly infected sources (i.e. T_{in}) as a function of time. In Figure 3, we plot the average time to infect T_{in} (see Eq.3) as a function of the number of infected hosts. The horizontal line shows the detection time (T_d) for a /8 telescope of a worm instance with an average scanning rate of 350 scans per second. Note that since we are concerned with detecting scans from unique infected hosts, T_d remains constant regardless of the worm's progress. T_{in}, on the other hand, varies as a function of the worm's propagation. The dots on Figure 3 depicts the time to infect for a simulated worm with the parameters listed in Table 2. While the simulation shows some variability (attributable to the randomness in node selection), it is evident that the analytical model accurately predicts the overall inter-arrival trend.

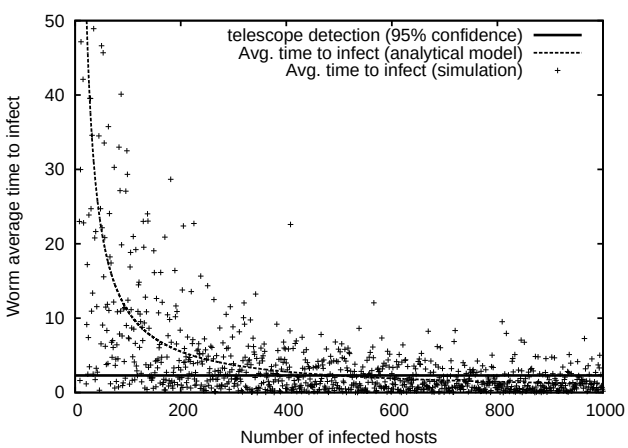

Figure 3: T_{in} as a function of the simulation parameters given in Table 2.

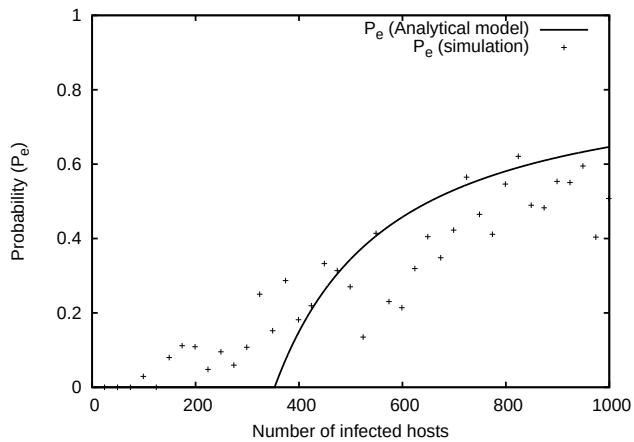

Figure 4: Probability that more than one host is infected *and* detected within T_d.

We show in Figure 4 how the probability P_e (cf. Eq.(8)) that more than one hosts is infected *and* detected within T_d changes as the number of infected hosts increases. As the graph illustrates, during the initial infection stages the probability that the worm infects any additional host within the telescope's detection time remains negligible. However, as the number of scanners increases, the time to infect a new host decreases and in doing so increases the likelihood of detecting more than one infected host within T_d. Not surprisingly, at this point the telescope's capability of inferring the actual worm evolution will start to deteriorate. We point out that P_e starts to grow in Figure 4 when about 400 hosts have been infected which is also the same point in Figure 3 when $T_{in} \approx T_d$.

3.2 Worm Evolution Similarity

We now evaluate the ability of telescopes to accurately estimate the evolution of a worm infection including the identification of patient zero. Specifically, we assess the sensitivity of the method presented in the previous section to the following factors: the size of the vulnerable population, the telescope size, and the worm scanning rate.

We view the actual worm evolution sequence as a sorted list of IP addresses $\mathcal{A} = \{a_1, a_2, \ldots, a_m\}$ where host a_i was infected before host a_j if $i < j$. Similarly, the sequence generated by the telescope is (a possibly different) set $\mathcal{B}$. The similarity of set $\mathcal{B}$ to set $\mathcal{A}$ can then be expressed as:

$$Y_{\mathcal{B}\to\mathcal{A}} = \sum_{i=0}^{m} \frac{\left(m - r_{(e_i,\mathcal{A})}\right)}{1 + |r_{(e_i,\mathcal{B})} - r_{(e_i,\mathcal{A})}|} \tag{9}$$

where, m is the maximum rank over which we compute the similarity between the two sets. e_i is the ith element in set $\mathcal{A}$ and $r_{(e_i,\mathcal{B})}$ is the rank of element e_i in set $\mathcal{B}$ created by the telescope.

A few points are worth mentioning regarding Eq. (9): First, by varying the maximum rank m we can calculate the similarity of the two lists for different subsets of $\mathcal{A}$ and $\mathcal{B}$ and therefore assess the accuracy of the telescope view at different stages of the infection. Second, $Y_{\mathcal{B}\to\mathcal{A}}$ assigns higher weight for elements existing in both sets that appear towards the initial stages of the infection. We do this since this period is critical in detecting the origins of the worm evolution. Finally, the similarity $Y_{\mathcal{B}\to\mathcal{A}}$ metric is penalized by the difference in ranks of the elements in set $\mathcal{B}$ and the actual set $\mathcal{A}$.

Next, we compute the similarity between the telescope generated sequence and that of the actual worm evolution, and show beyond which point the measure decreases in quality to be of practical use. To do so, we simulate the worm infection by initially selecting a single infected host at random (i.e., patient-zero), which in turn starts spreading the worm. The worm evolution sequence generated by the simulator is used as our baseline (that is, to derive set $\mathcal{A}$), simultaneously, we record the timestamps of the first scan from each unique infected host hitting a `/8` telescope. The sequence observed by the telescope represents the inferred worm evolution sequence, namely set $\mathcal{B}$. We then apply the metric given in Eq. (9) to compute the similarity between the two sets, and normalize the results by the similarity of set $\mathcal{A}$ with itself. The normalized similarity of the two sets is shown in Figure 5. Each point on the graph represents a similarity score (averaged over ten simulation runs) between the two sequences up to the first m infected host. Observe, for example, that up to the first 200 infected hosts, the `/8` telescope is capable of re-constructing the actual worm evolution with a similarity score of 0.98. Accuracy degrades below 0.75 after more than 4000 hosts have been infected.

3.2.1 *Effect of monitor size:*

To gain a better understanding of the reliability of the view generated by different monitor sizes, we consider the optimistic case of having an aggregate view from two known /8 telescopes — CAIDA's [2] telescope and the iSink [7] — compared to a less fortunate scenario of having only a single `/16` telescope. These results are also depicted in Figure 5. As expected, the larger the size of the telescope relative to the vulnerable population size the more accurate it is in constructing the actual infection sequence, and can do so even farther into the progress of the worm propagation. For example, in the optimistic case, we retain a similarity score of above 0.8, until close to the point at which 50% of the vulnerable population has already been infected. The `/16` telescope, on the other hand, fairs significantly worse in correctly classifying the sequence from the very beginning of the worm propagation.

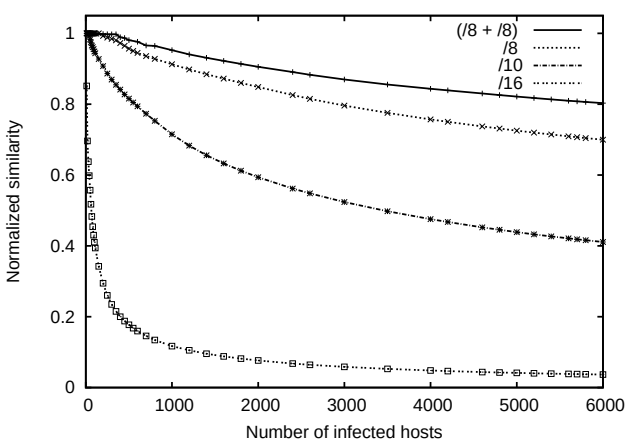

Figure 5: Normalized similarity of the telescope reconstructed sequence to the actual infection sequence of the worm for different telescope sizes.

3.2.2 *Effect of vulnerable population size:*

Our baseline case has a vulnerable population of only 12,000 hosts. In what follows, we consider the effect of larger vulnerable populations on the accuracy of the telescope, keeping all other parameters the same. As the results in Figure 6 show, the accuracy of the telescope is highly sensitive to the vulnerable population; higher population sizes cause T_{in} to drop very quickly below T_d, thereby increasing the likelihood of mis-ordering. In this case, had the vulnerable population been on the order of 100,000 hosts, the reconstruction correctness of the `/8` telescope would be severely limited beyond the first 450 infected hosts.

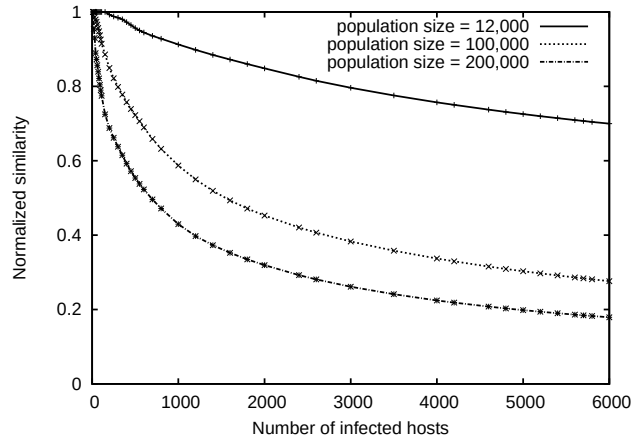

Figure 6: Normalized Similarity of the telescope reconstructed sequence to the actual infection sequence of the worm for different vulnerable population sizes.

3.2.3 *Effect of Scanning Rate:*

While we have shown that even at 350 scans per second telescopes of reasonable size can be successfully used to infer the actual infection evolution (at least during the early stages), we are also interested in evaluating the impact of scanning rate on this accuracy. In particular, we consider two aspects, namely, *(i)* the impact of varying the scanning rate and *(ii)* the impact of scanning rate inhomogeneity which has been observed particularly for bandwidth limited worms (e.g., UDP worms like Witty [15]).

In the first case, we evaluate the accuracy with scanning rates 10, 100, 500, and 1000 scans per second while keeping other simulation parameters unchanged. Somewhat surprisingly, we find that the scanning rate has minimal impact on the overall accuracy of the monitor view. This seems non intuitive at first, but on closer inspection notice that T_d and T_{in} both change with $1/s$. Hence, increasing the scan rate not only results in closer inter-arrivals of infected hosts, but also faster detection of a single scanner within T_d. For this reason, the telescope's accuracy is not significantly affected by uniform changes in the scanning rate.

In the second scenario, rather than considering a constant per-host scanning rate, for each worm instance we choose a randomly generated scanning rate. We investigate a case where this rate follows a normal distribution with average scanning rate of $\mu = 350$ scans/sec and standard deviation[3] $\sigma = 50$, and another where scanning rates are randomly generated following the heavy tailed distribution of Witty worm rates observed by CAIDA's telescope [15].

Figure 7 shows that the accuracy of the telescope view is significantly influenced by the inhomogeneity in scanning rates and drops sharply beyond the first 200 infected hosts. However, observe that

[3]We avoid generating non-positive scanning rates.

for all cases the telescope still provides an accurate estimate of the set of early infectees, which is arguably one of the most important aspects of worm evolution tracking.

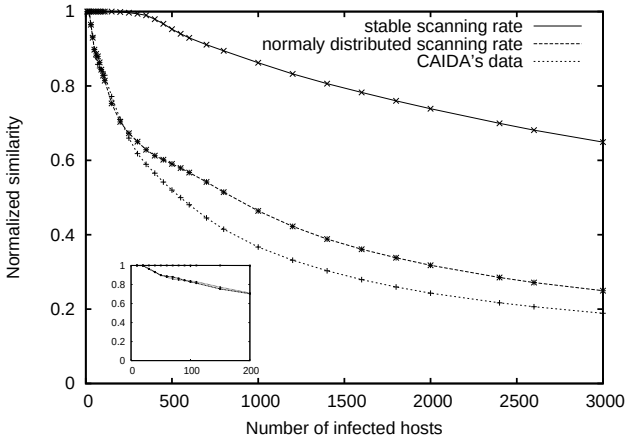

Figure 7: Normalized Similarity of the telescope reconstructed sequence to the actual evolution sequence of a worm with inhomogeneous scanning rate.

3.2.4 Effect of Packet Loss:

Packet loss presents yet another potential factor that can distort the telescope's view of the infection. As the population of infected hosts grows, the amount of probe traffic can exceed the network's capacity. Therefore scan packets may be dropped before they reach the telescope, hindering its view of newly infected hosts.

To investigate the effect of worm-induced packet loss on the accuracy of our method, we enhanced our simulator with the ability to mimic loss patterns resulting from worm propagation. Specifically, packet loss grows linearly with the number of infected hosts scanning the network. Experimental results showed that packet loss becomes an issue only after an overwhelming number of hosts have been infected. At this point however, $T_{in} < T_d$ and the accuracy of the telescope's view is already compromised.

4. DETECTING THE EXISTENCE AND SIZE OF THE HIT LIST

In Section 3 we assumed that the worm outbreak started with a single infected host. However, it is common for worm authors to instead use either a hit-list (i.e., a list of known vulnerable hosts) or previously compromised hosts to initiate the infection. In this case, identifying *patient zero* could be rather troublesome (and arguably infeasible beyond mere speculation) especially if the hit-list was targeted by a flash style targeted attack. However, we show that even in this case telescopes are still useful in detecting the existence of the hit list, its members, and its approximate size.

The intuition behind our approach is rather straightforward and is based on the fact that the hit-list is targeted by an out-of-band mechanism, and so its impact on the telescope is distinct from the normal scanning activity of the worm. In essence, the inter-arrival pattern of unique hosts from that list at the telescope manifests different characteristics compared to the arrivals of hosts infected by the worm's normal spreading behavior. In a flash style directed hit-list attack, one would expect that the inter-arrival times of the first scan from sources in the hit-list would be clustered in a relatively short interval with a nearly fixed average inter-arrival time at the telescope. Beyond the hit-list boundaries, however, the infected population exhibits the classical exponential growth which is distinct from that of the initial hit-list. This change in the inter-arrival pattern at the onset of worm spreading provides an indication of the existence of a hit-list, and by examining the boundary where the change occurs, one can unmask the size and members of that list.

We argue that this pattern will still persist under a wide range of parameters. To see why, consider for a moment that the worm starts by infecting a hit-list of size h_0 at time t_0. Given the telescope's single host detection capability, T_d, and assuming a uniform scanning rate, the average inter-arrival time of the first scan from hosts in the hit-list will be T_d/h_0. Then a hit-list with size h_0 can not be detected *if* the worm is able to infect new hosts such that:

$$T_{in} \leq \frac{T_d}{h_0} \tag{10}$$

Substituting for T_{in} and T_d from Eq.(3) and Eq.(5) we have:

$$\log\left(1 - \frac{1}{V - h_0}\right) \leq \frac{\log(1-\alpha)\ \log\left(1 - \frac{1}{2^{32}}\right)}{\log\left(1 - \frac{M}{2^{32}}\right)}$$

Solving the above equation for h_0 for a Witty-like worm, we found that given a `/8` telescope there is *no* such h_0 that satisfies the above inequality. This provides evidence that in the case of Witty, if a hit-list exists it will be detected by the telescope.

To illustrate how the change in the inter-arrival pattern leads to identifying the initial hit-list, we seed the simulator with an initial hit-list of 100 sources which are simultaneously infected at time zero. Figure 8 depicts the running average of the inter-arrival times of unique infected hosts as seen at the telescope — the stark contrast in inter-arrivals as we approach the boundary of the first 100 infected hosts reveals that the worm was initiated using a hit-list of approximately 100 hosts.

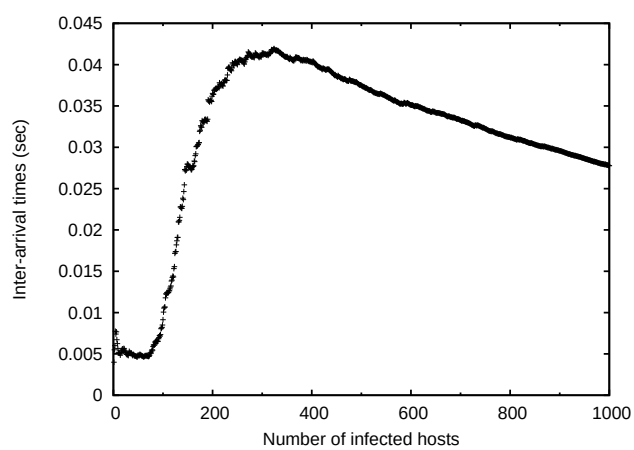

Figure 8: Average Inter-arrival time of a simulated worm with a hit list of size 100 hosts. The sharp change in the pattern of inter-arrivals at the boundary of the hit-list not only reveals the presence of a hit list, but also allows us to estimate its size.

To investigate whether this pattern is general, we explore the impact of different vulnerable population and hit-list sizes, as well as the impact of scanning rate inhomogeneity as illustrated in Section 3.2. To do so, we increase the vulnerable population to 50,000 hosts and choose an initial hit-list of 1,000 sources. Figure 9 shows

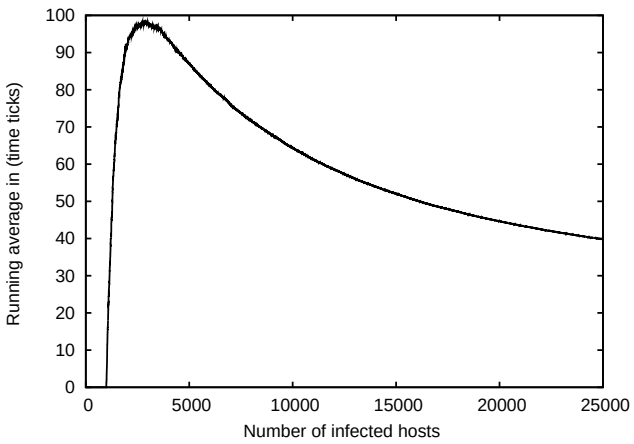

Figure 9: Average Inter-arrival time of unique infected hosts as seen by a /8 telescope for a population size of 50,000 hosts and 1000 hosts in the hit list. The clear change in the pattern of inter-arrivals at the boundary of the hit list reveals that the worm used an initial hit list of approximately 1000 hosts.

the moving average of the inter-arrivals of unique sources at a /8 telescope. Because a larger hit-list is used, the change in the pattern of inter-arrivals is even more pronounced. The reason is that since more hosts are infected within the initial T_d, the inter-arrival time of sources at the onset of the worm is even smaller. For small hit list sizes, a similar experiment (not shown) was conducted with a hit-list of 10 sources, and found that the source inter-arrivals exhibit the same pattern at the boundaries of the hit-list. Additionally, we generated random scanning rates following the distribution observed by CAIDA's telescope [15] and the results showed that the change in inter-arrival pattern is still preserved at the boundaries of the hit-list even under highly non-homogeneous scanning rates.

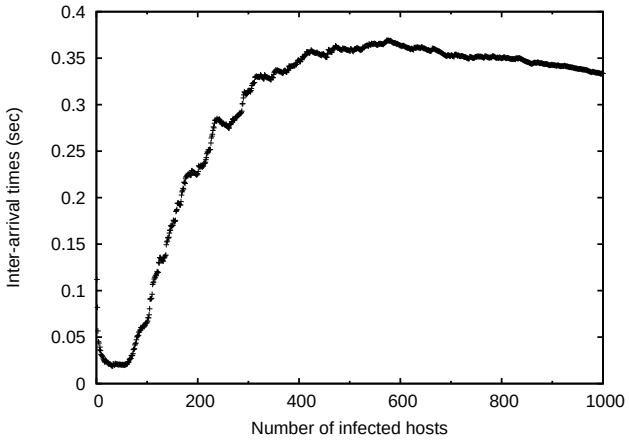

Figure 10: Average Inter-arrival time of unique infected hosts as seen by CAIDA's /8 telescope. Again, the clear change in the pattern of inter-arrivals at the boundary of the hit-list reveals that the Witty worm used an initial hit-list of approximately 80 hosts.

Finally, we validate our approach by studying the inter-arrivals of new Witty infected sources as observed by CAIDA's telescope. Figure 10 illustrates the running average of the inter-arrivals of the first 1,000 infected hosts. The graph reflects the same pattern observed in our simulations, with the change in behavior occurring near the first 80 infected hosts. Upon further inspection of the list of these sources, we find that 66 of the sources fall in the same `/16` subnet. Since Witty does not apply local preferential scanning, this clearly indicates that these hosts were infected by some other means. Moreover, reverse DNS lookups reveals that 70 sources in the list belong to the same institution, which strengthens the belief [15] that the worm was initiated by a targeted attack against known vulnerable hosts in that institution.

5. RELATED WORK

Over the last few years several researchers have used traffic monitors on unused address space (also called *network telescopes* or *traffic sinks*) to monitor large scale network security events and to provide forensic analysis of different network anomalies. Moore *et. al*, for example, investigated the prevalence of DoS attacks by analyzing incoming backscatter traffic to CAIDA's telescope [13] and provided analysis of pertinent aspects of global worm outbreaks [10, 12]. Baily *et. al* presented a distributed network monitoring and data collection infrastructure, called the Internet Motion Sensor (IMS) [1], and provided insights into various scanning activities by aggregating the views of theses monitors. Recently, Rajab *et. al* [14] explored a number of constraints related to distributed telescope deployment, and showed how such telescopes—if deployed correctly—can be used to detect non-uniform scanning worms early in the worms' propagation.

Forensic analysis of worms and understanding how they propagate is by no means a new problem. CAIDA [2] has provided detailed insights into different aspects pertaining to popular worm outbreaks. Recently, Shannon *et. al* [15] presented one such analysis for the Witty worm as detected by CAIDA's `/8` telescope. Of the elements presented there, were the worm spreading time, the vulnerable population size and the domains that they belonged to. Additionally, based on the observation that 110 of the hosts hitting the telescope arrived within the first 10 seconds of the outbreak—which clearly could not have occurred based solely on Witty's regular spreading dynamics (i.e., given its scanning rate and the size of the vulnerable population) — it was assumed that these hosts represented the initial hit-list. We analytically show that the existence of a hit list will result in unique inter-arrival pattern that changes at the hit-list boundaries, and showed how this pattern reveals the existence of the hit-list and it size. For the Witty case we arrive at a result similar to that presented in [15], though with a smaller hit-list.

Xie *et. al.* [23] proposed an algorithm to track the origin(s) of a worm infection. The algorithm uses random walks to sample edges in the connection history of hosts to deduce the worm causal tree of infection which, as the authors showed, is rooted at the first host that starts the infection at the monitored network. For the case of multiple sources starting the infection at the same time (i.e. a hist list), the algorithm converges to the multiple entry points on the causal graph representing the initial hit-list. While the results are sound, the technique assumes full knowledge of all hosts contact graphs over an unlimited time range. By contrast, the approach we present is simpler in nature since it only requires having traces collected at the network telescope, and does not assume any knowledge of the topology or hosts connection history.

Lately, Kumar *et.al.* [8] presented a forensic analysis of the Witty worm by reverse engineering the worm and exploiting several flaws in the random number generator. The authors show how this in-depth analysis can reveal some fairly interesting aspects pertaining

to worm spreading (e.g. the IP space missed by the Witty worms scans, the up-time of a number of machines, the number of disks on the infected hosts, etc). By correlating different events the authors speculated the identity of *patient zero*. While this result still needs to be validated, one would argue whether the aberrant events that are relied upon indeed pinpoint patient zero, particularly given the presence of a hit list. Nonetheless, the authors provide interesting insights that hopefully can be applied to similar findings for other malware events.

6. SUMMARY

Reconstructing the course of a worm's infection has been an interesting challenge. In this paper we presented a simple technique that uses the history of scans from unique hosts as seen by a network telescope to infer the actual sequence of host infections. While such an approach appears simplistic at first, our analysis verified by simulation results shows that it can very accurately track the initial (and most crucial) steps of an infection. The effectiveness of our technique deteriorates earlier in the progress of the worm propagation, for smaller telescope sizes, or worms with highly non-homogeneous scanning rates, or for large vulnerable population sizes relative to the monitor size. However, in several cases the telescope is still able to reconstruct the worm evolution especially towards the early stage of its propagation with reasonable accuracy.

We have also shown how we can exploit changes in the pattern of unique source inter-arrivals at the telescope to detect the existence and infer the size of a hit-list. We show this pattern persists over a wide range of hit-list and vulnerable population sizes. Finally, we validated our mechanism for identifying the hit-list using the inter-arrivals of new Witty infected sources as observed by CAIDA's telescope.

Acknowledgments

This work is supported in part by National Science Foundation grant SCI-0334108. We thank our shepherd, Stuart Staniford, for his suggestions on ways to improve this paper. We also extend our gratitude to the anonymous reviewers for their insightful comments.

7. REFERENCES

[1] Michael Bailey, Evan Cooke, Farnam Jahanian, Jose Nazario, and David Watson. Internet motion sensor: A distributed blackhole monitoring system. In *Proceedings of the ISOC Network and Distributed System Security Symposium (NDSS)*, 2005.

[2] CAIDA. http://www.caida.org.

[3] CERT. CERT Advisory CA-2001-26 Nimda Worm. http://www.cert.org/advisories/ca-2001-26.html.

[4] CERT. Code Red II: Another Worm Exploiting Buffer Overflow in IIS Indexing Service DLL. http://www.cert.org/incident_notes/in-2001-09.html.

[5] Zesheng Chen, Lixin Gao, and Kevin Kwiat. Modeling the Spread of Active Worms. In *Proceedings of IEEE INFOCOMM*, volume 3, pages 1890 – 1900, 2003.

[6] H.W. Hethcote. The Mathematics of Infectious Diseases. In *SIAM Reviews, Vol. 42 No. 4*, 2000.

[7] iSink- University of Wisconsin. http://wail.cs.wisc.edu/.

[8] Abhishek Kumar, Vern Paxson, and Nicholas Weaver. Exploiting Underlying Structure for Detailed Reconstruction of an Internet-scale Event . In *Technical Report*.

[9] David Moore. Network Telescopes: Observing Small or Distant Security Events. In 11th *USENIX Security Symposium, Invited Talk*, August 2002.

[10] David Moore, Vern Paxson, Stefan Savage, Colleen Shannon, Stuart Staniford, and Nicholas Weaver. Inside the Slammer Worm. *IEEE Magazine of Security and Privacy Magazine*, pages 33–39, July 2003.

[11] David Moore, Colleen Shannon, Geoffrey M. Voelker, and Stefan Savage. Internet Quarantine: Requirements for Containing Self-Propagating Code. In *Proceedings of IEEE INFOCOM*, 2003.

[12] David Moore, Collen Shannon, and Jeffry Brown. Code-Red: A case study on the spread and victims of an Internet worm. In *Proceedings of Internet Measurement Workshop*, pages 273–284, November 2002.

[13] David Moore, Geoffrey M. Voelker, and Stefan Savage. Inferring Internet Denial of Service Activity. In *Proceedings of* 10th *USENIX Security Symposium*, August 2001.

[14] Moheeb Abu Rajab, Fabian Monrose, and Andreas Terzis. On the Effectiveness of Distributed Worm Monitoring. In *Proceedings of the* 14th *USENIX Security Symposium*, pages 225–237, August 2005.

[15] Colleen Shannon and David Moore. The Spread of the Witty Worm. *IEEE Security and Privacy Magazine*, 2(4):46–50, July 2004.

[16] Stuart Staniford. Containment of Scanning Worms in Enterprose Networks. In *Journal of Computer Security*, 2004.

[17] Stuart Staniford, Vern Paxson, and Nicholas Weaver. How to 0wn the internet in your spare time. In *Proceedings of the* 11th *USENIX Security Symposium*, August 2002.

[18] Jamie Twycross and Matthew M. Williamson. Implementing and Testing a Virus Throttle. In *Proceedings of the 12th USENIX Security Symposium*, August 2003.

[19] Nicholas Weaver, Vern Paxson, Stuart Staniford, and R. Cunningham. A taxonomy of computer worms. In *Proceedings of the First ACM Workshop on Rapid Malcode (WORM)*, 2003.

[20] Nicholas Weaver, Stuart Staniford, and Vern Paxson. Very Fast Containment of Scanning Worms. In *Proceedings of the 13th USENIX Security Symposium*, August 2004.

[21] Matthew M. Williamson. Throttling Viruses: Restricting Propagation to Defeat Malicious Mobile Code. In *Proceedings of the 18th Annual Computer Security Conference*, pages 61–68, December 2002.

[22] The CAIDA Dataset on the Witty Worm - March 19-24, 2004, Colleen Shannon and David Moore, http://www.caida.org/passive/witty/. Support for the Witty Worm dataset and the UCSD Network Telescope are provided by Cisco Systems, Limelight Networks, DHS, NSF, CAIDA, DARPA, Digital Envoy, and CAIDA Members.

[23] Yinglian Xie, Vyas Sckar, David A. Maltz, Michael K. Reiter, and Hui Zhang. Worm Origin Identification Using Random Moonwalks. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 242–256, May 2005.

[24] Cliff Zou, Weibo Gong, and Don Towsley. Code Red Worm Propagation Modeling and Analysis. In *Proceedings of ACM Conference on Computer and Communication Security (CCS)*, pages 138–147, 2002.