A Multi-Resolution Approach for Worm Detection and Containment*

Vyas Sekar vyass@cs.cmu.edu
Yinglian Xie ylxie@cs.cmu.edu
Michael K. Reiter reiter@cmu.edu
Hui Zhang hzhang@cs.cmu.edu
Carnegie Mellon University

Abstract

Despite the proliferation of detection and containment techniques in the worm defense literature, simple threshold-based methods remain the most widely deployed and most popular approach among practitioners. This popularity arises out of the simplistic appeal, ease of use, and independence from attack-specific properties such as scanning strategies and signatures. However, such approaches have known limitations: they either fail to detect low-rate attacks or incur very high false positive rates. We propose a multi-resolution approach to enhance the power of threshold-based detection and rate-limiting techniques. Using such an approach we can not only detect fast attacks with low latency, but also discover low-rate attacks – several orders of magnitude less aggressive than today's fast propagating attacks – with low false positive rates. We also outline a multi-resolution rate limiting mechanism for throttling the number of new connections a host can make, to contain the spread of worms. Our trace analysis and simulation experiments demonstrate the benefits of a multi-resolution approach for worm defense.

1. Introduction

Worms pose a significant threat to the dependability of existing and future networking infrastructure. Defending against such self-propagating attacks in an automated fashion is a challenging task, and has sparked much interest in the research community. Existing approaches for worm defense (e.g., [3, 7, 13, 18]) have been shown to be effective for very fast, non-polymorphic, random scanning worms. However, they leave open to attackers opportunities to circumvent the defense mechanisms by exploiting the very assumptions used for detection and containment. Future attacks can evade detection mechanisms which depend on scanning rates, signatures, and other attack-specific features.

Interestingly, one of the earliest known scan-detection heuristics, threshold-based detection based on the number of unique destinations contacted, is applicable across a wide spectrum of worm attacks. Threshold-based mechanisms are very popular and are one of the most widely-deployed worm defenses [12] due to their simplicity and ease of deployment. The strength and robustness of the mechanism lies in its minimal set of assumptions about the nature of attacks – scanners contact many unique destinations. By adopting a metric which is invariant across scanning attacks, independent of the scanning strategy and content signatures, this approach has the ability to be *attack-agnostic*.

However, threshold-based detection mechanisms currently lack the accuracy and effectiveness of attack-specific approaches. Having only a single fixed threshold for a metric (such as the number of unique destination addresses contacted), typically measured over a single time window a few seconds long, network administrators must make a choice in the selection of the detection threshold. The choice is between a high threshold that can detect only very high-rate attacks but has low false positive rates, and a low threshold that can detect low-rate stealthy attacks but that may have a very high false positive rate. This fundamental inflexibility limits the practical applicability of threshold-based approaches to high-rate attacks. A natural question is: can we retain the attack-agnostic properties of threshold-based detection, but provide detection capabilities comparable to attack-specific approaches?

Our solution is a *multi-resolution* approach for detecting and containing worms, without depending on attack-specific scanning properties and signatures. The key insight behind the multi-resolution approach is the following simple yet powerful observation. While the short term connection patterns of normal end-hosts may

*This work was partially supported by NSF grant number CNS-0433540, and by KISA and MIC of Korea.

be bursty, involving a large volume of traffic and connections to many unique destination addresses, hosts exhibit lower average connection rates when observed over longer timescales. As a result, we find that connection metrics, such as the traffic volume and the number of distinct destinations contacted, grow as a *concave* function of the size of the time window (i.e., the second derivative with respect to the time window size is negative). This suggests that using multiple resolutions with different detection thresholds at different time granularities will be an effective solution to detect a wide range of attack rates with low false positive rates.

Our traffic analysis (Section 3) confirms this intuition, and indicates the potential benefits of a multi-resolution approach. We provide a systematic framework (Section 4) for realizing these benefits, by balancing the inherent tradeoff between the false positive rate and the detection latency (and hence the potential damage caused by infected hosts). We define the security cost of a system, in terms of the false positive rates and detection latencies, and outline an optimization framework for selecting parameters optimally for a multi-resolution detection system.

Our multi-resolution approach for containment (Section 5) draws upon a similar insight in the nature of end-host behavior. Locality in destination address selection suggests that throttling connections to new destinations that have not been contacted previously, will achieve the desired containment capability without disrupting the activity of normal hosts. Our evaluations demonstrate that a multi-resolution approach achieves enhanced containment capabilities over traditional approaches.

2. Related Work

Prior work has focused on understanding different worm propagation models (e.g., [10, 16]). Many techniques have been proposed for detecting worm outbreaks using either large-scale monitoring infrastructures (e.g., [19]) or locally deployed honeypots (e.g., [5]). There are also several systems for efficient and fast worm signature generation (e.g., [3, 7, 14]).

There has been surprisingly little work on detection of stealthy, low-rate, scanning attacks. Staniford et al. describe a mechanism for detecting stealthy port scans [15] arising outside the network, by using a historical probability model for different types of traffic. Our work focuses on detecting and throttling infected hosts inside a local network similar to [9, 13]. There are two compelling reasons for deploying such capabilities. First, rate limiting can reduce wasteful bandwidth consumption and avoid overloading network and router resources. Second, such approaches can curb the internal spread of worms that exploit topological locality.

Chen and Tang [2] propose worm detection and containment based on connection failure rates. Jung et al. use sequential hypothesis testing [6, 13] to detect scanners by tracking failed connection attempts. Our approach is agnostic to the scanning strategy since it does not rely on failed connections.

Several worm containment methods have been suggested in the literature, including rate-limiting, quarantine, and signature-based filtering. Moore et al. [11] study the limits on the responsiveness of content-filtering and address-blacklisting as containment measures, while Wong et al. [18] discuss the effectiveness of rate-limiting mechanisms. Zou et al. [20] present an analytical framework for reasoning about worm propagation in the presence of defense mechanisms. Williamson proposed the virus throttle [17] based on the observation that the number of connections to previously uncontacted hosts is fairly low. While the class of containment measures we evaluate have been proposed earlier in these contexts, our contribution is the design and evaluation of a multi-resolution approach for rate limiting.

Multi-resolution analysis in spatial and temporal dimensions, using Fourier and wavelet analysis, has been suggested for anomaly detection (e.g., [1, 4]). Calculating the number of unique destinations contacted over multiple time resolutions necessarily involves taking *unions* of the set of destinations contacted over multiple time bins. Signal analysis techniques are not suitable in this context as they cannot capture the semantics of such a union operation for multi-resolution analysis.

3. Motivation

In threshold based anomaly detection, the traffic monitor identifies abnormal activity by measuring specific traffic metrics and flagging suspicious observations which exceed a pre-set threshold within a specific time window. Commonly used metrics for detecting abnormal host behavior include the total traffic volume (number of packets or flows) and the number of unique destination addresses contacted (regardless of whether the connection was successful or not). Despite their widespread deployment, threshold-based mechanisms suffer from an inherent inflexibility arising from the conflicting goals in threshold selection. A large (i.e, conservative) threshold that accounts for normal traffic bursts will not be able to detect low-rate attacks, while a small (i.e., aggressive) threshold will result in high false

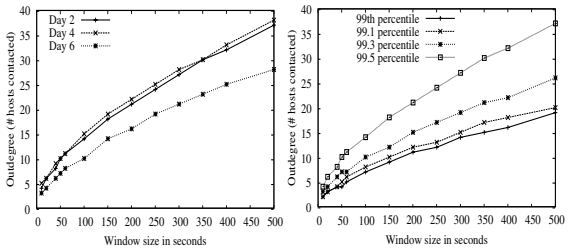
positive rates where even small bursts of legitimate activity will be flagged as potentially anomalous.

With respect to worm detection, the metric of interest is the number of unique destination addresses contacted. If the number of unique destination addresses contacted by a benign host grows as a linear function of the time window, then a single-resolution approach operating with a fixed threshold is sufficient, as it will uniquely identify the (malicious) scanning rates we can detect. There are two observations which suggest that the number of unique destinations contacted will grow slower than a linear function of the time-window size. First, while normal traffic can be very bursty at short timescales, such bursts are seldom sustained for a longer period of time. Second, there is a significant amount of locality [8, 17] in the connection patterns of end-hosts. A host is likely to "talk" to destinations it has contacted before, and the number of new destinations contacted is low. If the growth trend (as a function of the time window size) is *concave*, i.e., the second derivative is less than or equal to zero,[1] then a single-resolution approach may no longer be sufficient.

Dataset Description: We confirm this intuition regarding the nature of end-host behavior, using a week-long packet-header trace collected between September 28 and October 4, 2003 at the border router of a university department. The router observes all traffic between internal hosts and the rest of the Internet (including other university hosts, file servers, and mail servers). The traces were anonymized, by removing packet payloads, and anonymizing IP addresses using a prefix-preserving anonymization scheme.[2]

In our analysis, we assume that each unique valid IP address inside the network corresponds to a unique end-host. This assumption is valid as there is no NAT/DHCP usage within the department. Due to the possibility of scans to invalid addresses, and the lack of information on the IP ranges used, we use the following heuristic for identifying valid addresses from the anonymized trace. First, we identified the most significant 16 bits of internal IP addresses space (after anonymization) of hosts within the network. If a host from within this known /16 network prefix successfully completed a TCP handshake with an external host (i.e., outside the /16), we select the host for analysis. Using this heuristic, we identified a set of 1,133 valid addresses in the week-long trace.

Traffic Analysis: For each of the 1133 identified hosts, we use the following method to measure the number of distinct destinations it contacts. For TCP connections, we identify the packets with the `SYN` flag set, and add the destination to the *contact set* of the source. For UDP connections, we use a flow-based approach to identify the directionality of session initiation, i.e., the host which sends the first packet in a UDP session (with a timeout of 300 seconds) is considered the flow initiator, and we add the destination of this flow to the contact set of the source. We repeated our analysis with an undirected notion of connectivity (without session initiation semantics) and observed similar results. For the remainder of this paper, we use only the directional notion of connectivity.



(a) Growth of 99.5th percentile for different days

(b) Growth of different statistics for Day 2

Figure 1. Traffic growth is concave, suggesting a multi-resolution approach

The trace was binned into $T = 10$ second non-overlapping intervals, and we computed the number of distinct destination addresses contacted by each identified host over different window sizes using these binned observations. For our analysis, we used time window sizes ranging from 20 seconds to 500 seconds (i.e., from 2 to 50 bins). Given a window of size w seconds, we consider all possible sliding windows consisting of w/T bins. The number of destinations contacted by a host within the window of size w seconds will then be the union of the set of hosts contacted across w/T consecutive bins, each of duration T seconds.

Figure 1(a) shows the growth of the 99.5th percentile of the number of distinct destinations contacted for three different days in the week-long trace. Analyzing the slope of the curve we find that the growth trend as a function of the time window size is indeed concave. Figure 1(b) shows the growth of different statistical percentiles of the observed traffic on the second day of the trace. We observe that the concave trends are consistent across different statistical percentiles as well.

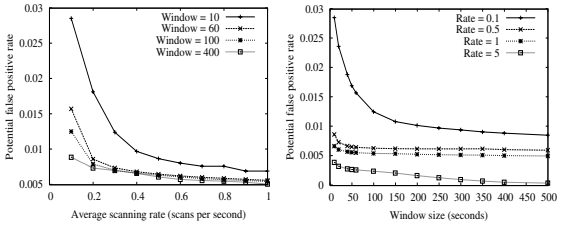
Next we proceed to analyze the detection capabilities of different time resolutions for different worm attacks. We characterize an attack in terms of the

[1] While the growth may show convex behavior temporarily over small time ranges, it suffices if the overall (macro) behavior is concave.

[2] `tcpdriv`, `http://ita.ee.lbl.gov/html/contrib/tcpdpriv.html`.

rate r, which is the number of unique destination addresses contacted by each infected host per second. With a single-resolution threshold-based scheme, to detect worms with rate greater than r scans per second using a window of size w seconds, we would choose a detection threshold (for the number of unique destinations contacted) to be equal to $r \times w$. For a fixed rate r, a threshold lower than $r \times w$ always detects the worm, and a higher threshold never does. Thus the notion of a false negative rate for detection of a fixed worm-rate r is not relevant for threshold-based detection. Hence, we focus on the potential false-positive rates of using different window sizes, for detecting different worm-rates. The false-positive rate for detecting worm-rate r at window size w is the probability that a normal host contacts more than $r \times w$ unique destination IPs within a w second window. Using the week-long trace, we obtain a conservative[3] estimate of the false positive rate by calculating the number of events where one of the 1133 hosts within our network exceeds the connection threshold of $r \times w$ unique destinations contacted within a w second sliding window.

Figure 2 shows the false positive rates using two different views – one fixing w and varying r, and the other fixing r and varying w. We note that the false positive rates decrease with larger time windows, suggesting that the resolution window can be a tunable parameter to tradeoff false positive rate and detection latency.



(a) Varying worm-rate (b) Varying window size

Figure 2. Analyzing false positive rates

Such trends suggest that we can simultaneously use different threshold values, each applied at a different time resolution, to detect a wide range of attack rates. This is the key idea behind a multi-resolution approach. Using multiple thresholds at multiple time windows, we should be able to detect fast attacks at small time windows, and low rate attacks at larger time windows. A multi-resolution approach can ensure not only low detection latency for fast scanning attacks, but also provide a new capability for exposing stealthy scanning attacks, both with low false positive rates. Thus we can detect a wide spectrum of attack rates, independent of signatures and scanning strategies, while retaining the ease of use of threshold-based approaches.

[3]This is a conservative estimate since we might be detecting real scanning activity as well.

4. Multi-Resolution Detection

The measurement study indicates the potential benefits of using a multi-resolution approach. Figure 3 depicts the various steps involved in the systematic design to realize these benefits. The first step involves identifying traffic metrics of interest for anomaly detection and rate-limiting. We use the number of unique destinations contacted, since it is largely independent of worm-specific properties.

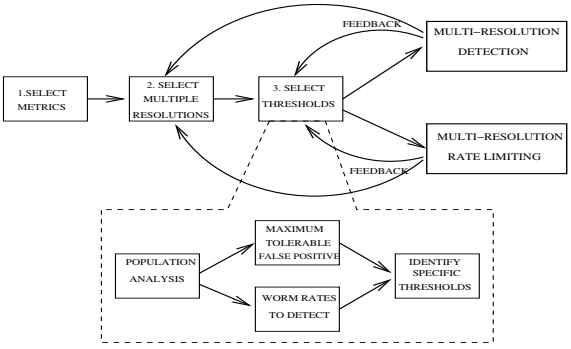


Figure 3. Design of a system for multi-resolution detection and containment.

The next two steps involves the identification of different window sizes and deriving detection thresholds for each different window. The threshold selection step can be viewed as an optimization procedure, which given the operating costs and constraints specified by the network administrator, selects detection thresholds for the different window sizes optimally. This process is guided by historical traffic profiles of the host population. Over time, administrators can provide additional feedback to fine-tune the system parameters, using deployment-specific expertise.

4.1. Threshold Selection

This section outlines the threshold selection procedure for a multi-resolution detection system, delineating the set of tradeoffs and constraints involved.

Input:

- The desired detection capability of the system, specified by a range of worm-rates $R = [r^{min}, r^{max}]$. As a simplifying assumption, we assume that R is a discrete set consisting of all values

between r^{min} and r^{max}, in increments of a predefined step value r^{step}.
- The set of time resolutions W, between w^{min} and w^{max}, over which end-host behavior is monitored.
- The third input to the formulation is a set of different $fp(r_i, w_j)$ values, for each $r_i \in R$ and $w_j \in W$, where $fp(r_i, w_j)$ is the false positive rate associated with identifying the worm rate r_i using a time resolution of window size w_j. Since we adopt a data-driven approach for parameter selection, we assume that each administrator has historical traffic profiles of hosts within their network. For a given r_i and w_j, $fp(r_i, w_j)$ can be obtained from the historical traffic profiles, by computing the number of hosts that contacted greater than $r_i \times w_j$ unique destinations in w_j seconds (similar to the analysis in Section 3).

Section 4.4 suggests guidelines on how network administrators can select these input parameters.

Security Cost: There are two orthogonal objective functions in the design of a detection system: the false positive rate and the potential damage done by the attack before detection. If we wanted the lowest possible false positive rate and did not care about the damage done by worms with scanning rates in the spectrum of rates specified by R, then we would use a single-resolution approach using the largest window-size in W (w^{max}) with a threshold corresponding to $r^{min} \times w^{max}$. On the other hand, if the only concern is with respect to the damage caused, then a single-resolution approach using the smallest window-size in W (w^{min}) with a threshold of $r^{min} \times w^{min}$ would be optimal. There is an inherent tradeoff between the false positive rate and the damage done, and the goal is to find an optimal multi-resolution approach in this design space.

We formalize these two orthogonal cost factors as the *Detection Accuracy Cost* (DAC), which is a function of the false positive rate, and the *Detection Latency Cost* (DLC), which is a function of the total damage that an attack causes before it is detected. For worm and scanning attacks, a natural notion of the DLC is in terms of the number of destination addresses an infected host contacts before it is detected as an anomaly.

The security cost is a function of the DAC and the DLC. The goal of this paper is not to construct an ideal cost model for intrusion detection systems. Rather, we wish to demonstrate the potential benefits of a multi-resolution approach. Hence, to model the security cost of the detection system, we use a simple linear combination, $Cost = DLC + \beta \times DAC$. The parameter β (specified by the network administrator) needs to account for the possibly different scales over which the two cost factors are measured, and to possibly normalize the DAC and the DLC into a uniform dollar-cost. Intuitively, β lets us achieve the desired tradeoff between latency and accuracy. Administrators who want a conservative detection system (i.e., lower false positive rate) would select a high β, while those who desire a more aggressive detection approach (i.e., lower detection latency) would select a lower β.

Objective: For a given β, the goal is to minimize the security cost $Cost = DLC + \beta \times DAC$, i.e., to find detection thresholds for the different time windows which minimizes the overall security cost of the system.

ILP Framework: We present an Integer Linear Programming (ILP) formulation modeling cost criteria and detection constraints.

First, we define $\{0,1\}$ variables δ_{ij}, to model the assignment of different worm-rates to different windows.

$$\delta_{ij} = \begin{cases} 1 & \text{if rate } r_i \text{ is assigned to time-window } w_j \\ 0 & \text{otherwise} \end{cases}$$

To represent the fact that each worm rate has to be assigned to some time window, i.e., the system must detect all rates within the desired spectrum, we have the following detection constraints:

$$\forall i, \sum_{j=1}^{|W|} \delta_{ij} = 1$$

The false positive rate (f_i) associated with the detection of worm-rate r_i can be expressed in terms of the different *fp* values available from the historic traffic profiles. Since the rate r_i is assigned to exactly one of the windows, we have:

$$f_i = \sum_{j=1}^{|W|} fp(r_i, w_j) \times \delta_{ij}$$

The damage done by the worm rate r_i before it is detected, can be written as:

$$d_i = \sum_{j=1}^{|W|} r_i \times w_j \times \delta_{ij}$$

The latency cost DLC, caused by the set of worm-rates R, is expressed as:

$$DLC = \sum_{i=1}^{|R|} d_i - d_i^{min}$$

Here d_i^{min} represents the damage done if we use the smallest available time window in the set W for detection, i.e., $d_i^{min} = r_i \times w^{min}$. The DLC models the additional damage that is allowed by possibly choosing a longer detection latency for each worm-rate.

The last part of the formulation is to find an expression for the false positive cost criterion DAC, as a function of the individual f_i values. The overall false positive rate of a multi-resolution system cannot be expressed directly as a combination of the false positive rates of individual resolutions, i.e., we cannot obtain an analytical closed form due to possible overlap across alarms from different time resolutions. For example, a fast scanning host may be flagged as anomalous by both the smaller window sizes and by the larger window sizes, even though it is conceptually a single alarm.

It appears that unless we try out every possible combination of time-resolutions and thresholds, we cannot obtain the DAC. Instead of using a brute-force approach of trying all possible combinations, we present two simple alternative models: *Conservative* and *Optimistic*, that can be used in the formulation.

For the conservative combination, we take the DAC to be the sum of the individual false positive rates. The conservative model assumes that there is no overlap between the alarms from different time resolutions, and hence adds up the false positive rate across all the worm rates. Formally,

$$DAC_{Conservative} = \sum_{i=1}^{|R|} f_i$$

For the optimistic combination, we take the DAC to be the maximum of the individual false positive rates. The optimistic estimate assumes that the alarms across the different time resolutions overlap completely, and as a result the overall false alarm cost will be the maximum across the different worm-rates. This can be formally expressed using the following linear constraints:

$$\forall i, DAC_{Optimistic} \geq f_i$$

Output: The above formulation[4] can then be solved to obtain the optimal δ_{ij} assignments. Given the different δ_{ij} values, the thresholds for the multi-resolution approach are easy to obtain. For each window-size w_j, with at least one δ_{ij} being non-zero, the threshold is $r_j^{min} \times w_j$, where r_j^{min} is the smallest worm-rate assigned to w_j.

4.2. Analysis

We select R, with $r^{min} = 0.1$ scans/second, in increments of $r^{step} = 0.1$, up to $r^{max} = 5$ scans/second. For W, we use a minimum time window of 10 seconds, and a maximum time window of 500 seconds. The *fp* estimates are obtained as described in Section 3. We use a general purpose constraint optimization solver `glpsol` to obtain the optimal assignments for the different δ_{ij} values, for both the conservative and optimistic models. Obtaining the optimal solution with `glpsol` was fairly efficient – within one second with 50 worm-rates and 13 window sizes. We observe that for the conservative DAC model, a simple greedy algorithm can provide the optimal assignments. Each worm rate r_i is assigned to the window size $w^*(i)$ that minimizes the function $r_i \times w_j + \beta \times fp(r_i, w_j)$. It is easy to see why the greedy assignment is optimal. If in the optimal solution, r_i is assigned to $w_j \neq w^*(i)$, changing the assignment of r_i to $w^*(i)$ will only reduce the contribution of r_i to the overall cost, and hence will reduce the overall cost.

[4]We have found that in noisy datasets it is necessary to add constraints so that thresholds increase monotonically with window size.

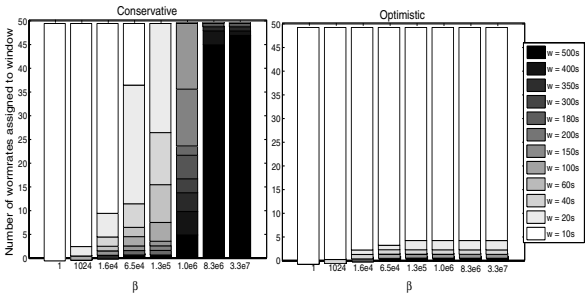


Figure 4. Contributions of different time resolutions for different β values.

To study the tradeoff between the DAC and the DLC, we vary β. Figure 4 shows the number of worm rates assigned to each window size as a function of β. This helps visualize the contribution of the different resolutions in a multi-resolution approach. With low β we expect that the latency factor dominates, and a majority of the rates will be assigned to the smaller time-windows. As β increases, the contribution of the false positive cost becomes more dominant and the assignment will tend to distribute more evenly across the time windows. We also observe that for large values of β the DAC dominates, causing the assignment to be completely biased toward the largest window size (500 seconds in our analysis). Due to the nature of the optimistic cost model, we find that the distribution is rather skewed, and only a small number of time resolutions (4-5) are used at any given time. With the conservative model, we observe that the assignments are more evenly distributed.

4.3. Implementation

A multi-resolution detection system can be deployed at the access and internal routers of an enterprise, either as a stand-alone system or as a module in popular

```
MULTIRESOLUTIONDETECTION(W, T, H, M)
    // W is the set of time resolutions
    // T : W → ℝ is the set of thresholds
    // H is the set of hosts
    // M : H × W → ℝ is the set of measurements
1   for each host h ∈ |H| do
2       for each window w ∈ |W| do
            // Check if it exceeds the threshold
3           if (M(h, w) > T(w))
            then
4               A(h) ← 1
5       if (A(h) > 0)
        then
            // t is the current timestamp
6           Flag ⟨h, t⟩ as an anomaly
```

Figure 5. Multi-resolution detection

IDSes (e.g., [12]). We have implemented a proof-of-concept prototype of the multi-resolution detection system, which monitors the network activity of each internal host using multiple resolutions. Our current implementation is a stand-alone version, running on a commodity desktop (Pentium IV 2.4 GHz, 1 GB RAM), emulating a real-time detection system by reading in a packet trace through a `libpcap` front-end. Even with very few code optimizations in our implementation, the CPU and memory requirements for performing such multi-resolution detection in a network with over a thousand hosts are small, suggesting that such an approach is feasible for small to medium size enterprise networks.

The procedure for multi-resolution detection is outlined in Figure 5. The detection system first obtains the number of distinct destination addresses contacted by each host (in the set H) using sliding windows of different sizes (in the set W) to obtain the set M of per-host measurements. $T(w)$ represents the threshold for the number of unique destinations contacted as a function of the time window w. These thresholds are obtained from the output of the ILP framework described in Section 4.1. For each host h, and each window size w, we check if the measured value is greater than the detection threshold $T(w)$. A host's behavior is flagged as anomalous if its activity exceeds the threshold for at least one of the constituent resolutions, i.e., conceptually we are taking the union of the alarms raised in each of the window-sizes. Each alarm raised by the system is a tuple of the form $\langle hostid, timestamp \rangle$, which means that $hostid$ exceeded the connection threshold for at least one of the time windows ending at $timestamp$.

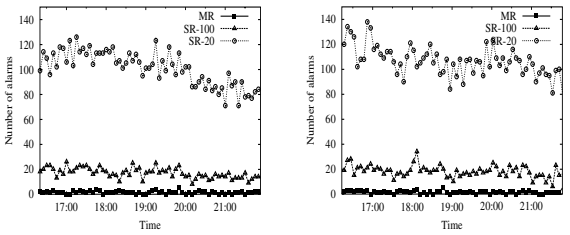
We evaluate our prototype using traces collected on two additional days (Oct 8th and 9th, 2003) as test data to evaluate the potential false alarm rates. The threshold settings were derived from the same input settings used in Section 4.2, using a conservative cost model with $\beta = 65536$. As described in Section 3, we bin the data into 10 second bins, and for each bin we get the set of unique destination IP addresses contacted by each of the 1133 hosts within the network.

We found it useful to include a reporting mechanism that coalesces alarms temporally. The temporal aggregation allows us to report a single alarm for anomalies which are localized in time, instead of generating an alarm for each anomalous observation. The temporal clustering procedure identifies the start and end of an alarm event, and clusters together anomalous observations for a given host that are close in time. For example, if for a given host we have alarms at times $t_i, t_{i+1}, .., t_{i+k_1}$ and $t_j, t_{j+1}, .., t_{j+k_2}$, with $j > i + k_1 + 1$, we report it as only two alarms at times t_i and t_j instead of generating $k_1 + k_2$ alarms.

Figure 6 shows the alarms generated by alternative approaches for specific snapshots on the two different days. For purposes of visualization, we aggregate alarms over five minute time intervals, and show only a four-hour snapshot. Table 1 summarizes the number of alarms of a multi-resolution detection approach and single-resolution approaches of different window sizes. The multi-resolution approach is denoted as *MR*, and a single-resolution approach using a window of size w is denoted as *SR-w*. The thresholds for the single-resolution approaches are chosen to be able to detect all possible worm rates that the multi-resolution approach can detect. From our results, we observe that the number of alarms generated with single-resolution approaches is up to two orders of magnitude greater than the multi-resolution approach.

Analyzing the alarms from the multi-resolution approach, we found that more than 65% of the alarms are raised by less than 2% of the hosts in the network. This suggests that the effective workload of a system administrator to investigate these alarms will be significantly less than the number of alarms raised. Further, the number of alarms generated by our system is not unmanageable for manual or semi-automated diagnosis, and the alarm rates reported above are in fact conservative estimates of the actual false-positive rate.[5] From these observations, we believe our multi-resolution approach serves as a practical starting point for detection of stealthy, low-rate scans.

[5] Due to trace anonymization and absence of payload information, we could not independently verify true positives within the alarms.



(a) Oct 8, 2003 (b) Oct 9, 2003

Figure 6. Comparing multi-resolution and single-resolution detection

Detection Approach	Number of alarms (per 10-seconds)			
	Oct 8		Oct 9	
	Average	Maximum	Average	Maximum
SR-20	3.37	16	3.19	18
SR-100	0.56	6	0.53	8
SR-200	0.17	5	0.15	5
MR	0.04	2	0.04	2

Table 1. Summary of alarms

4.4 Discussion

Section 4.1 assumes that the network administrator provides R and W values for obtaining the detection thresholds. R would be selected based on the range of worm-rates the administrator is interested in, i.e., the desired detection capability of the system. The choice of W depends on the computation and memory resources available. The memory requirement is determined by w^{max}, the largest window size in W, while the compute load depends on the number of windows chosen (i.e., $|W|$). Having a wider spectrum of W and more fine-grained selection of window sizes can only improve the threshold selection. If using a small subset of W gives a solution with better security cost, the optimization framework will automatically use only these useful window sizes. Our experiments indicate that even simple and coarse-grained selection of W and R yields substantial performance benefits.

For the *fp* values, it would be desirable that they are obtained from "clean" historical traffic profiles. Since obtaining completely noise-free traffic data is not practical, in our evaluations we find the use of conservative false positive estimates (i.e., treating possible true positives as potential false positives) to be a reasonable approximation for threshold selection. The reason is that the effect of a small number of true positives and isolated data anomalies on a large population distribution is rather minimal. With larger population sizes and lengths of historical traffic profiles, the effect of data anomalies can be further minimized.

Section 4.1 finds threshold settings that minimize the security cost for detecting a given spectrum of worm-rates. Alternatively, the administrator may desire to maximize the spectrum of worm-rates that can be detected with a multi-resolution detection system, for a given constraint on the operating cost. This can in fact be obtained through a process of iterative refinement which uses our ILP formulation as a sub-routine. The administrator can start with $r^{min} = 0$, obtain the minimal security cost from the ILP solver, and check if the parameters returned meet the cost constraints. If the constraints are not met, then she can adaptively refine R by increasing r^{min}, until the security cost meets the operating cost constraint.

5 Multi-Resolution Rate-Limiting

While detection of infected hosts may help in faster deployment of patches or generation of worm signatures, it does not have a direct impact on attack containment. In this section, we describe and evaluate a multi-resolution approach to worm containment, where we rate limit the number of distinct destinations that an infected host connects to.

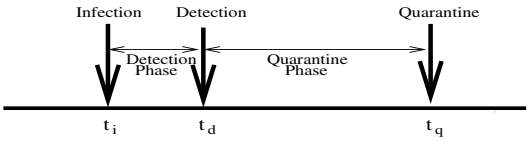


Figure 7. Timeline of an infected host

There are two phases during which an infected host is active (Figure 7), the *detection* phase (before its activity raises an alarm) and the *quarantine* phase (before it stops generating more malicious traffic into the network). Quarantine typically involves manual or semi-automated investigation of the detected host by a network administrator. The administrator can subsequently quarantine the infected host either by isolating it from the network, or by "cleaning" it and applying vulnerability-specific patches. While the damage caused by an infected host during the detection phase (between t_i and t_d) is unavoidable since it takes a non-zero amount of time to discern malicious activity, the damage inflicted during the quarantine phase (t_d to t_q) can be minimized using rate limiting mechanisms.

Our multi-resolution approach to rate limiting is described in Figure 8. The input to the rate limiting module is a set of time resolutions W and connection limiting thresholds T, in terms of the number of (unique) destinations that a host is allowed to contact. Let t_d^h denote the detection time for host h. Suppose at time t

MULTIRESOLUTIONCONTAINMENT(W, T)

// W is the set of time-windows
// $T : W \rightarrow \mathbb{R}$ is the set of containment thresholds
1 Detect possibly anomalous hosts H
2 **for** each flagged host $h \in H$ **do**
3 Let t_d^h be time at which host h was flagged
// Suppose h attempts to contact x at time t
// Find the nearest, higher time window
4 $Upper_{t-t_d^h} \leftarrow \min_{w \in W} w \geq (t - t_d^h)$
// AC is number of connections allowed
5 $AC \leftarrow T(Upper_{t-t_d^h})$
// CS is the Contact Set, initially empty
6 **if** $(|CS(h)| > AC)$
then
7 Deny this connection
else
8 Allow connection, Add x to $CS(h)$

Figure 8. Multi-resolution containment

($> t_d^h$), host h attempts to contact destination x. If x is already in h's contact set, the connection is allowed. If x is not in the contact set, then the rate limiting mechanism checks if this connection can be allowed, i.e., whether h has exceeded its connection threshold for the next higher time window. If the threshold has already been exceeded the connection is denied, otherwise the connection is allowed and x is added to h's contact set.

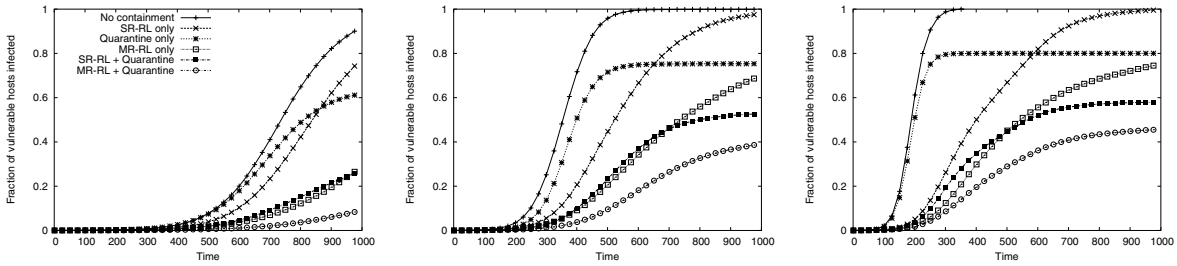
To evaluate the effectiveness of a multi-resolution approach for rate limiting we performed simulation experiments, emulating the spread of a random scanning worm attack over a host population of size $N = 100000$ hosts. We assume that the total address space is twice the size of the host population, and set the fraction of hosts vulnerable to five percent. We model the duration of the quarantine phase ($t_q - t_d$ from Figure 7) as being uniformly distributed between 60 and 500 seconds. We use the multi-resolution detection system, described in Section 4.3, as our anomaly detection mechanism. The length of the detection phase will thus be the smallest time window at which an infected host exceeds its connection threshold.

We compare the containment capabilities of a multi-resolution rate-limiting approach against a single-resolution approach. For the multi-resolution approach we use the same set of windows used in the detection module (Section 4.3), while for the single-resolution approach we use a window of size 20 seconds. To perform a fair comparison across the two rate-limiting approaches, we need to select throttling thresholds such that the overall false positive rates (i.e., disruption caused to normal connections) are normalized. We choose the thresholds for multi-resolution and single-resolution rate-limiting to be equal to the 99.5th percentile of the traffic distributions at different window-sizes (described in Section 3). This ensures a fair comparison, since it normalizes the false positive rates of both methods to be $100 - 99.5 = 0.5\%$.

There are six combinations of quarantine and rate limiting mechanisms. At one extreme, we have a worm spreading with no containment mechanisms in place, and at the other extreme we have multi-resolution rate limiting used in conjunction with quarantine. Figure 9 shows the growth of the different scanning worms, in terms of the fraction of vulnerable hosts that have been infected as a function of time. Each simulation experiment was repeated over 20 independent runs, and we report the average over the 20 runs. We find that across all three scanning rates, the multi-resolution rate limiting (MR-RL) mechanism outperforms the single-resolution rate limiting (SR-RL) and quarantine-based containment measures. For example, with a scanning rate of 0.5 scans/second, we find that the fraction of vulnerable hosts infected at time $t = 1000$ seconds, with MR-RL+Quarantine is only 10%, which is one-third of the fraction of hosts infected with SR-RL+Quarantine, and just one-sixth of the fraction of hosts infected using quarantine alone. Across different scanning rates, we find that the multi-resolution approach gives at least a two-fold improvement over a single-resolution approach. In fact, we notice that the containment effect of MR-RL is comparable to that of SR-RL and quarantine used together.

6 Conclusions

The multi-resolution approach presented in this paper tackles the notion of stealthy scanning attacks along the dimension of scanning rate, providing the ability to detect and contain attacks across a wide spectrum of scanning rates. By focusing on an *attack-agnostic* metric, the number of distinct destinations contacted by a host, our approach is robust across a large class of worm attacks as it is independent of scanning strategies and worm signatures. The simple but powerful observation that guided our design is that this traffic metric grows as a concave function of time. Our experiments show that such an approach significantly enhances the detection and containment capabilities against a wide spectrum of slow propagating worm attacks. As future work, we are adding more spatial and temporal traffic profiles, and other relevant traffic metrics into the multi-resolution framework.



(a) 0.5 scans/second (b) 1 scan/second (c) 2 scans/second

Figure 9. Comparing different containment mechanisms for various worm scanning rates

Acknowledgments

We thank Chenxi Wang and Stan Bielski for providing us access to the packet traces used in our analysis. We also thank David Maltz, David Brumley, and the anonymous reviewers for their comments which helped improve the paper.

References

[1] P. Barford, J. Kline, D. Plonka, and A. Ron. A Signal Analysis of Network Traffic Anomalies. In *Proceedings of ACM SIGCOMM Internet Measurement Workshop (IMW)*, 2002.

[2] S. Chen and Y. Tang. Slowing Down Internet Worms. In *Proceedings of IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2004.

[3] M. Costa, J. Crowcroft, M. Castro, A. Rowstron, L. Zhou, and P. Barham. Vigilante: End-to-End Containment of Internet Worms. In *Proceedings of ACM Symposium on Operating System Principles (SOSP)*, 2005.

[4] M. Crovella and E. Kolaczyk. Graph Wavelets for Spatial Traffic Analysis. In *Proceedings of IEEE INFOCOM*, 2003.

[5] D. Dagon, X. Qin, G. Gu, W. Lee, J. Grizzard, J. Levin, and H. Owen. HoneyStat: Local Worm Detection Using Honeypots. In *Proceedings of Symposium on Recent Advances in Intrusion Detection (RAID)*, 2004.

[6] J. Jung, V. Paxson, A. W. Berger, and H. Balakrishnan. Fast Portscan Detection Using Sequential Hypothesis Testing. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2004.

[7] H. Kim and B. Karp. Autograph: Toward Automated, Distributed Worm Signature Detection. In *Proceedings of USENIX Security Symposium*, 2004.

[8] J. McHugh and C. Gates. Locality: A New Paradigm for Thinking About Normal Behavior and Outsider Threat. In *Proceedings of the New Security Paradigms Workshop (NSPW)*, 2003.

[9] J. Mirkovic, G. Prier, and P. Reiher. Attacking DDoS at the Source. In *Proceedings of IEEE International Conference on Network Protocols (ICNP)*, 2002.

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

[11] D. Moore, C. Shannon, G. M. Voelker, and S. Savage. Internet Quarantine: Requirements for Containing Self-Propagating Code. In *Proceedings of IEEE INFOCOM*, 2003.

[12] M. Roesch. Snort - Lightweight Intrusion Detection for Networks. In *Proceedings of USENIX Large Installation System Administration Conference (LISA)*, 1999.

[13] S. E. Schechter, J. Jung, and A. W. Berger. Fast Detection of Scanning Worm Infections. In *Proceedings of Symposium on Recent Advances in Intrusion Detection (RAID)*, 2004.

[14] S. Singh, C. Estan, G. Varghese, and S. Savage. Automated Worm Fingerprinting. In *Proceedings of USENIX/ACM Symposium on Operating Systems Design and Implementation (OSDI)*, 2004.

[15] S. Staniford, J. A. Hoagland, and J. M. McAlerney. Practical Automated Detection of Stealthy Portscans. *Journal of Computer Security*, 10:105–136, 2002.

[16] S. Staniford, V. Paxson, and N. Weaver. How to 0wn the Internet in Your Spare Time. In *Proceedings of USENIX Security Symposium*, 2002.

[17] M. M. Williamson. Design, Implementation and Test of an Email Virus Throttle. In *Proceedings of Annual Computer Security Applications Conference (ACSAC)*, 2003.

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

[19] J. Wu, S. Vangala, L. Gao, and K. Kwiat. An Effective Architecture and Algorithm for Detecting Worms with Various Scan Techniques. In *Proceedings of Networking and Distributed Systems Security Symposium (NDSS)*, 2004.

[20] C. C. Zou, W. Gong, and D. Towsley. Worm Propagation Modeling and Analysis under Dynamic Quarantine Defense. In *Proceedings of ACM CCS Workshop on Rapid Malcode (WORM)*, 2003.