Distributed Evasive Scan Techniques and Countermeasures

Min Gyung Kang, Juan Caballero, Dawn Song

February 9, 2007
CMU-CyLab-07-004

CyLab
Carnegie Mellon University
Pittsburgh, PA 15213

Distributed Evasive Scan Techniques and Countermeasures

Min Gyung Kang, Juan Caballero, Dawn Song

Carnegie Mellon University,
{mgkang,jcaballero,dawnsong}@cmu.edu

Abstract. Scan detection and suppression methods are an important means for preventing the disclosure of network information to attackers. However, despite the importance of limiting the information obtained by the attacker, and the wide availability of such scan detection methods, there has been very little research on the evasive scan techniques, which can potentially be used by attackers to avoid detection. In this paper, we first present a novel classification of scan detection methods based on their amnesty policy, since attackers usually take advantage of such policies to evade scan detection methods. Then we propose two novel metrics to measure the resources that an attacker needs to complete a scan without being detected. Following, we introduce *z-Scan*, a novel evasive scan technique that uses distributed scanning, and show that it is extremely effective against TRW, one of the state-of-the-art scan detection methods. Finally, we investigate possible countermeasures including hybrid scan detection methods and information-hiding techniques. We provide theoretical analysis, as well as simulation results, to quantitatively measure the effectiveness of the evasive scan techniques and the countermeasures.

Keywords: scan detection, evasion, distributed scanning, information-hiding

1 Introduction

Network scans have become a common and useful means for hackers to obtain information on a specific network, such as detecting active hosts and ports in service [3] or as a tool for reconnaissance before attacking the vulnerable hosts. In an effort to detect and prevent these scan activities, various scan detection methods have been proposed [11, 12, 14, 15, 18–22]. These *scan detection* methods have been widely deployed, often in combination with *scan suppression* methods that try to limit the information obtained by the attacker. Typically, the output of the scan detection method becomes one input to the scan suppression method. For example, the scan detection method may output the IP address of a remote host performing a scan on the local network. Then, the suppression method takes care of blocking any further traffic from that address.

However, despite the importance of limiting the information obtained by the attacker, and the wide availability of scan detection methods, there has been very little research on the *evasive scan techniques* that can potentially be used by attackers to avoid detection. Moreover, the metaphor for security co-evolution, “security arms race”, is

also true for this case as attackers develop new evasive scan techniques to elude scan detection methods. Multiple techniques have been invented for this purpose such as *dumb scan* [1], distributed scan, and several stealthy port scan techniques [2]. Thus, it is imperative to analyze evasive scan techniques and explore countermeasures against them.

In this paper, we make the following contributions:

Classify scan detection methods according to their amnesty policy: Scan detection methods assign anomaly scores to a host's activities. As this score will ever increase, they use an amnesty policy to lessen scores in the case of normal activities. These amnesty policies usually constitute a vector for evasive scan techniques and thus need to be properly studied. We present a novel classification for scan detection methods based on their amnesty policy: *Positive-Reward*-based and *Timeout*-based methods. Such a classification allows us to abstract the essence of these scan detection methods and facilitates the analysis of evasive scan techniques against each family.

Propose two new evaluation metrics: Scan detection methods have been mostly evaluated with respect to their accuracy and detection delay. We propose using two additional metrics to evaluate scan detection methods to incorporate the notion of how many resources the attacker needs to complete the scan, in the presence of that scan detection method, and yet remain undetected. That is, how easy it is to obtain the information while evading that scan detection method. The metrics are: 1) the time that it takes an attacker to complete the scan of a network and 2) the number of IP addresses that the attacker needs to complete the scan. In both cases assuming the presence of the scan detection method and that the attacker wishes to remain undetected.

Introduce z-Scan, a new evasive scan technique: We introduce *z-Scan*, a new evasive distributed scan technique against Positive-Reward-based methods. In particular, we show z-Scan to be effective against Threshold Random Walk (TRW), which has been shown to be one of the most effective scan detection methods in terms of speed and accuracy. Our z-Scan technique is extremely effective against TRW; it can scan without being detected a given address space, protected with TRW, using a small number of source addresses, which is bounded logarithmically with respect to the size of the address space.

Propose a hybrid solution to z-Scan: We propose using a hybrid scan detection method that combines Positive-Reward-based and Timeout-based detection to defend against evasive scan techniques. Through our analysis, we demonstrate that Positive-Reward-based detection methods and Timeout-based detection methods are synergistic and when combined, can be much more effective at defending against evasive scan techniques.

Analyze information-hiding as a solution to z-Scan: We explore information-hiding techniques as another type of countermeasure against evasive scan techniques. Information-hiding techniques, rather than trying to detect and block scans, try to hide the true information about the network, and hence reduce the utility of the scans. Through theoretical analysis and simulation results, we show that information-hiding based countermea-

sures are promising against evasive scan techniques; in particular, in the case of TRW, this can completely render z-Scan ineffective.

The rest of the paper is organized as follows. In Section 2, we classify scan detection methods according to their amnesty policy and introduce the metrics we use to evaluate them. In Section 3 we introduce and analyze z-Scan, our new evasive scan technique. Then, in Section 4, we propose a hybrid scan detection method to counter z-Scan, and provide theoretical analysis and evaluation results on its effectiveness. In Section 5, we analyze the effectiveness of other countermeasures against z-Scan based on information-hiding techniques. Finally, we discuss related work in Section 6 and conclude in Section 7.

2 Classification of Scan Detection Methods & Evaluation Metrics

In this section, first we introduce a novel classification of scan detection methods based on their amnesty policy and then we propose new metrics that can be used to evaluate the effectiveness of a scan detection method when faced with evasive scan techniques.

2.1 Classification of Scan Detection Methods

There has been ample research on scan detection methods [12, 15, 18–20, 22]. However, all these methods are based on one common principle: if the accumulated score for a host's activities exceeds a certain threshold value, the host is considered a scanner. As this accumulated score will ever increase and eventually hit the threshold, detection methods usually provide policies to lessen the scores in the case of normal activities, which we call *amnesty policies*.

Such policies are important, because attackers try to scan stealthily and amnesty policies, if exploited maliciously, can provide a way for an attacker to make its behavior look normal. As such, amnesty policies are a likely vector for evasive scan techniques and we need to understand how these policies work and how they can be exploited. As a first step, we propose a novel classification of scan detection methods based on their amnesty policy, which yields three categories: *(1) Positive-Reward-based* methods; *(2) Timeout-based* methods; and *(3) No Amnesty* methods.

Positive-Reward-based Methods Positive-Reward-based methods lessen the accumulated anomaly score upon the occurrence of normal events such as successful connection attempts or connections to highly visited hosts. Threshold Random Walk (TRW) [12] and its variants [20, 22] as well as Leckie et al.'s probabilistic approach [14] fall into this category.

TRW uses a random walk to decide whether a new connection initiated by a host is benign or malicious. We explain it in detail in Section 3 but simply put, it keeps a ratio for each host and in the case of a successful connection started by that host, multiplies its ratio by a value less than 1, making the ratio farther from a fixed threshold (and

vice versa in the case of failed connection attempt). Leckie et al assign anomaly scores to probes, based on the access probability for each target host and thus connections to highly visited hosts are considered normal.

Timeout-based Methods Timeout-based methods assign a lifetime to each event. The lifetime is decreased periodically (i.e. events age) and events expire when their lifetime is expired. Thus, the amnesty policy is based on expiration of events. Events that have been expired are no longer used to compute the anomaly score for each host. These methods can be again categorized into two groups according to their methods for assigning a lifetime to each event:

- Uniform lifetime (Block Scan Detection)
  The methods in this class count the number of events (or sum of the anomaly scores for all events) contained in a fixed time window and check if a threshold is exceeded. Snort [19] counts the total number of connection attempts, while Kato et al. [13] consider only failed connection attempts. Basu et al.'s approach [8] uses neural networks to assess the score for each event and compares the sum of scores, during a fixed time window, with the threshold.
- Lifetime proportional to how anomalous the event is considered
  The Spice engine [21] grants each packet a lifetime proportional to its anomaly score [1] to impede evasion attacks using *delayed scan* techniques.

No Amnesty Methods There are a few traditional scan detection methods that don't provide any way to reduce the anomaly score of a host such as Bro[2] [15] and the Robertson et al. detection method [18]. However, as this accumulated score will ever increase and eventually hit the threshold (or several thresholds in the case of Bro), these scan detection methods are prone to false positives as any host, given enough time, will eventually be flagged as a scanner.

2.2 Evaluation Metrics for Scan Detection Methods

Evaluation metrics are needed to measure the effectiveness of scan detection methods. Although previous work has used different metrics to evaluate the effectiveness of a scan detection method, these metrics have been targeted to measuring the false positive rate, false negative rate and detection delay of the scan detection method.

In this work we propose two additional metrics that allow us to measure the effectiveness of a scan detection method under a specific scan technique used by an attacker. The idea behind these metrics is that the more resources the attacker needs to complete

[1] The Spice engine uses Bayes network to build up a profile for each source address.

[2] Here we refer to the classical scan detection method in scan.bro. Bro also has an option to employ the Threshold Random Walk (TRW) algorithm for enhanced scan detection and it has also recently introduced methods to evict state [10].

the scan in the presence of that detection method, while remaining undetected, the more difficult to evade the scan detection method is under that scan technique.

Time to complete the scan: The first metric we propose is the time that it takes for an attacker to complete the scan of a network in the presence of the scan detection method, using a specific scan technique. Note that this is different from measuring detection delay, since we are interested in how long it takes *the attacker* to complete the scan of a certain address space, rather than how long it takes the *scan detection method* to detect an attacker.

Clearly, due to the frequent changes in the topology of any network (e.g. dynamic IP assignments, laptops, hosts being replaced, etc), the longer it takes for an attacker to complete the scan the less truthful the information gathered at the beginning of the scan is compared to the current topology. Also, the topological information gathered by the attacker, can have a lifetime after which it becomes useless. For example after the public announcement of a vulnerability, an attacker might be interested in promptly locating all vulnerable hosts in a protected network to try to compromise them. The time window for the attacker to perform the scan and the following attack is the time needed by the system administrator to identify the vulnerability, download a patch (if available) and install it in all vulnerable hosts.

Number of addresses needed to complete the scan: The second metric is the number of IP addresses that the attacker needs to complete the scan in the presence of the scan detection method, using a specific scan technique. For example, for a Timeout-based method, given the attacker has no time constrains, the attacker will need a single IP address to complete the scan, as it just needs to use a scan rate below the lowest detected by the method. We call the evasive scan technique of probing at a rate below the minimum detected by a method, a *delayed scan*.

But when the attacker has a time constraint, it needs to increase the number of addresses performing the scan in parallel, if it wants to complete the scan satisfying the time constraint while remaining undetected. For a Positive-Reward-based method, given an attacker, which randomly probes addresses in the target network using a single address, the detection method will eventually flag the address as a scanner. If the network employs scan suppression, blocking any further probes from that address, then the attacker needs to use another address to continue with the scan, thus requiring multiple IP addresses to send scans from.

To summarize, given an IP address space to scan, we can evaluate the effectiveness of a scan detection method against a specific scan technique, i.e. the number of resources needed by the attacker to complete the scan while remaining undetected, using a tuple (α, T), where α denotes the number of source addresses needed to complete the scan and T is the time needed for the attacker to complete the scan.

3 z-Scan: Evasion Attacks against TRW

Positive-Reward-based methods lessen the accumulated anomaly scores for a scanner upon occurrence of benign events. Thus, they provide an opportunity to the intelligent

attacker for evading detection if it is able to replicate or forge the existence of such benign events. In this section we propose new evasive scan techniques to evade detection by Positive-Reward-based methods that decrease the anomaly score of a scanner based on successful connections. We focus on TRW [12] as a representative of this family.

TRW is a well accepted scan detection method mainly used for detecting horizontal scans, where an attacker probes multiple protected hosts to obtain information about which hosts/services are available in the protected network. It can be applied to detecting vertical scans, as well as detecting misbehaved hosts inside the protected network. In this paper for simplicity, we focus on horizontal scans, similar techniques can be applied to other cases.

We first show *naive scan* as a straw-man case, and then describe a more sophisticated evasive scan technique we call *z-Scan*, that is very effective against TRW. For both techniques we analytically compute the values of α, the number of IP addresses that the attacker needs to complete the scan. Table 1 shows the notation used in the analysis.

N: size of address space to scan (number of active hosts + inactive IP addresses)
a: the number of active IP addresses in the address space
P_s: fraction of active hosts. *i.e.,* $\frac{a}{N}$
α: number of source IP addresses for the attack to scan the entire address space
H_0: the hypothesis that the source is a benign user
H_1: the hypothesis that the source is a scanner
$\Lambda(Y)$: the likelihood ratio for TRW
θ_0: the probability that a connection attempt succeeds given the hypothesis H_0
θ_1: the probability that a connection attempt succeeds given the hypothesis H_1
η_1: the upper threshold of the likelihood ratio $\Lambda(Y)$ which if crossed, flags the source as a scanner
n: the number of probes that attackers can perform before being blocked
n_i: the number of probes which the attacker can perform before being blocked at i-th round
s_i: the estimated number of accumulated active known hosts at i-th round
t: threshold of Block Scan Detection method, *i.e.*, the number of failed connection attempts within the time window.
β: fraction of correct scan result
w: size of the time window in Block Scan Detection method (in time ticks)
T: time constraint within which scan should complete
r: probing rate (the number probes per time tick)

Table 1. Notation

3.1 Naive Scan against TRW

Naive scan A determined attacker who wants to complete the scan of a network, and controls a set of hosts (i.e. IP addresses), can perform what we call a *naive scan*. A naive scan is a distributed scan. In its most basic form the scan is performed sequentially. The

attacker selects one of the addresses it controls and starts scanning the target network. Assuming that the target network uses scan suppression, after several probes the address will be flagged as a scanner and further probes will be blocked. At that point the attacker selects a different scanner address and commands it to scan a new set of addresses, not yet scanned, until it gets blocked again. The process continues until the complete target address space has been scanned.

Note, that an attacker that wants to optimize the naive scan, rather than use its addresses sequentially can make them scan in parallel. This allows the attacker to reduce the time employed to complete the scan. Here, the attacker divides the target address space into disjoint subsets of addresses and assigns one such subset to a different scanner address under its control. The scanner addresses probe their corresponding subset until being blocked and report back to the attacker any target addresses that it could not scan before being blocked, so they can be assigned by the attacker to a different scanner, not yet blacklisted.

Analysis Here, we compute the number of distinct source IP addresses, α, needed to scan an address space of size N addresses, when a naive scan technique[3] is used against TRW.

Let H_0 be the hypothesis that the source of a connection attempt is a benign user; and let H_1 be the hypothesis that it is a scanner. TRW defines an indicator variable Y_i that represents the outcome of the first connection attempt from a scanner to an target host, where $Y_i = 0$ if the connection attempt was successful and $Y_i = 1$ if it failed. Each connection attempt regardless of its success is considered as an event.

Then conditional on the hypothesis H_0 and H_1, the TRW framework defines:

$$\begin{array}{ll} Pr[Y_i = 0|H_0] = \theta_0 & Pr[Y_i = 1|H_0] = 1 - \theta_0 \\ Pr[Y_i = 0|H_1] = \theta_1 & Pr[Y_i = 1|H_1] = 1 - \theta_1 \end{array}$$

that is, the parameters θ_0 and θ_1 represent the conditional probabilities of an event given the hypothesis H_0 and H_1.

TRW keeps a likelihood ratio $\Lambda(Y)$ for each scanner that has generated an event. For every successful connection the likelihood ratio is reduced by multiplying it by $\frac{\theta_1}{\theta_0}$, and for each unsuccessful connection the likelihood ratio is increased by multiplying it by $\frac{1-\theta_1}{1-\theta_0}$. If the likelihood ratio for a scanner address exceeds the upper threshold η_1, the address is flagged as a scanner. The reader can refer to [12] for a detailed explanation of the framework and how to set the associated parameters.

We assume that scan suppression is used in addition to TRW, so that any probes received from an address that has been determined by TRW to be a scanner are dropped.

Let n be the total number of events generated by the scanner address, that is, the total number of unique connection attempts to different target addresses, let s be the number of unique connection attempts that were successful, and let $n - s$ be the number

[3] In this case, we assume that the attacker selects a random target and sends a probing packet without any evasion technique.

of unique connection attempts that were unsuccessful. Then likelihood ratio for the scanner address is:

$$\Lambda(Y) = \Pi_{i=1}^{n} \frac{Pr[Y_i|H_1]}{Pr[Y_i|H_0]} = (\frac{\theta_1}{\theta_0})^s (\frac{1-\theta_1}{1-\theta_0})^{n-s} \quad (1)$$

In order to be flagged as a scanner the likelihood ratio for an address needs to exceed the upper threshold η_1, thus meeting the following condition:

$$(\frac{\theta_1}{\theta_0})^s (\frac{1-\theta_1}{1-\theta_0})^{n-s} \geq \eta_1 \quad (2)$$

For simplicity, we assume that the active hosts are uniformly distributed across the target address space and define P_s to be the fraction of active hosts in the address space having open the port that the attacker is using for the horizontal scan. Then, $s = nP_s$ and solving $(\frac{\theta_1}{\theta_0})^s (\frac{1-\theta_1}{1-\theta_0})^{n-s} \leq \eta_1$ for n we obtain:

$$n \leq \frac{\log \eta_1}{P_s \log \frac{\theta_1}{\theta_0} + (1-P_s) \log \frac{1-\theta_1}{1-\theta_0}} \quad (3)$$

Equation 3 shows an upper bound on the number of target addresses that a scanner address can probe before being detected.

For each scanner address used, the attacker is able to gain information about n new addresses, before the address is blocked by the scan suppression method. Thus, in a naive scan the number of source addresses needed by the attacker to complete the scan of the whole address space of size N, which we denote by α as stated in Section 2.2, is:

$$\alpha \geq \frac{N}{n} = N \frac{P_s \log \frac{\theta_1}{\theta_0} + (1-P_s) \log \frac{1-\theta_1}{1-\theta_0}}{\log \eta_1} \quad (4)$$

This result shows that the number of addresses that an attacker, using a naive scan technique, needs to completely scan a target network is bounded by a function which grows linearly with the size of the address space being scanned.

3.2 z-Scan against TRW

z-Scan Section 3.1 introduced a basic attacker that performed a distributed scan on a target network. In this section we propose a more intelligent attacker that takes advantage of the positive rewards awarded by TRW for successful connections. This attacker performs what we call a *z-Scan*.

A z-Scan is a distributed scan where the attacker uses each of its available scanner addresses to scan a subset of the target network. The main difference with the naive

scan is that in a z-Scan the set of scanners controlled by the attacker collude sharing the addresses of previously-found active hosts.

A known limitation of TRW is that if the attacker knows a set of active hosts in the target network, it can evade detection by alternating a random probe with a probe to a known active host, thus making the likelihood ratio oscillate without reaching the upper threshold η_1.

Assuming that the attacker has no information whatsoever about the target network at the beginning of the scan and that the target network once again uses scan suppression, the attacker proceeds as follows. First, the attacker selects one of its scanner addresses and performs random probing until the address becomes blocked. At that point, it commands the host owning that address to pass the set of active hosts found to another host which starts scanning alternating a known active host with a random probe till exhausting the set of known active hosts. Once that set is exhausted the new host continues with random probes until being blocked. The procedure is repeated until the complete address space of the target network has been scanned. Note that the attacker could also split the target address space into smaller spaces and perform z-Scan in parallel on them using multiple addresses.

We will refer to the sequence of scan probes from a scanner address before it gets blocked as a *round*. Intuitively, we can anticipate that the number of active hosts probed at each round will increase exponentially, thus bounding the value of α, the number of IP addresses needed to scan an address space of size N, logarithmically with respect to the address space size, N. This technique is named "*z-Scan*" because it zigzags its targets from known active hosts to unknown hosts and vice versa.

Analysis In round i, the attacker uses a scanner address to probe part of the target address space, using information gathered in all previous rounds. Let n_i denote the total number of probes to distinct target addresses performed by the scanner address in round i, and s_i denote the number of successful connections (or probes) to distinct target addresses performed by a scanner address in round i. Since we defined α to be the number of addresses needed by z-Scan to complete the scan of the target network, we know that $i \in [1, \alpha]$.

Note that the attacker begins probing with no known active hosts, then the number of probes the attacker can perform at the first round n_1 before being blocked, is given by Equation 3:

$$n_1 \leq \frac{\log \eta_1}{P_s \log \frac{\theta_1}{\theta_0} + (1 - P_s) \log \frac{1-\theta_1}{1-\theta_0}}$$

Thus, the number of active hosts found at the first round, $s_1 = P_s n_1$.

Since the attacker has yielded s_1 active host addresses, he can move to another source address and repeat this process until being blocked by TRW. At this second round, the attacker can employ s_1 known active hosts to alternate probing between the known active host addresses and the unknown, making TRW oscillate below the threshold, η_1. However, after consuming all the known active addresses, the attacker needs to

perform naive random probing. Therefore, we can find the accumulated number of active addresses by the second round, s_2, by solving the following two equations for s_2:

$$(\frac{\theta_1}{\theta_0})^{s_2}(\frac{1-\theta_1}{1-\theta_0})^{n_2-s_2} \geq \eta_1 \tag{5}$$

$$s_2 = s_1 + (n_2 - s_1)P_s \tag{6}$$

Therefore,

$$s_2 = (1 - \frac{P_s \log \frac{\theta_1}{\theta_0}}{P_s \log \frac{\theta_1}{\theta_0} + (1-P_s)\log \frac{1-\theta_1}{1-\theta_0}})s_1 + \frac{P_s \log \eta_1}{P_s \log \frac{\theta_1}{\theta_0} + (1-P_s)\log \frac{1-\theta_1}{1-\theta_0}}$$

Since $(1 - \frac{P_s \log \frac{\theta_1}{\theta_0}}{P_s \log \frac{\theta_1}{\theta_0} + (1-P_s)\log \frac{1-\theta_1}{1-\theta_0}})$ and $\frac{P_s \log \eta_1}{P_s \log \frac{\theta_1}{\theta_0} + (1-P_s)\log \frac{1-\theta_1}{1-\theta_0}}$ are constants, we can simplify s_2 by replacing them with k and l respectively. Hence:

$$s_2 = ks_1 + l$$

If the attacker repeats this process, the estimated number of accumulated active known hosts at i-th round, s_i, is:

$$s_i = ks_{i-1} + l \tag{7}$$

We can derive the single general form for s_i:

$$s_i = k^{i-1}(s_1 + \frac{l}{k-1}) - \frac{l}{k-1} \tag{8}$$

Let $m_i = n_i - s_{i-1}$, where m_i denotes the number of new IP addresses probed in the i-th round. We can compute m_i as $m_i = k^{i-2}(k-1)(s_1 + \frac{l}{k-1})\frac{1}{P_s}$.

By the last round, round α, the total number of new IP addresses probed in all the rounds should be N. Thus α is the smallest value such that

$$n_1 + \sum_{2 \leq i \leq \alpha} m_i \geq N$$

and solving for α, we obtain:

$$\alpha = \left\lceil \log_k \left(1 + \frac{P_s(N - n_1)}{s_1 + \frac{l}{k-1}} \right) \right\rceil + 1 \tag{9}$$

This result shows that the number of addresses that an attacker, using a z-Scan technique, needs to completely scan a target network is bounded by a function which grows logarithmically with the size of the address space being scanned.

As an example, we assume an attacker that uses z-Scan with the following parameters: $P_s = \theta_1 = 0.3$, $\theta_0 = 0.99$, $\eta_1 = \frac{P_D}{P_F} = \frac{0.99}{0.01}$ where P_D and P_F are desired detection probability and false positive probability as in [12]. If this attacker wants to scan a /8 network (i.e. 2^{24} addresses), then it only needs to use 110 addresses to complete the scan versus 9.5 million when using naive scan.

Figures 1 and 2 plot Equations 4 and 9 respectively using the above parameters. Compare the logarithmic bound of z-Scan with the linear bound of the naive scan. Clearly, an attacker that wants to avoid detection can take advantage of the positive reward method of TRW to limit the amount of resources (i.e. IP addresses) needed to complete the scan, which shows the vulnerability of TRW to z-Scan.

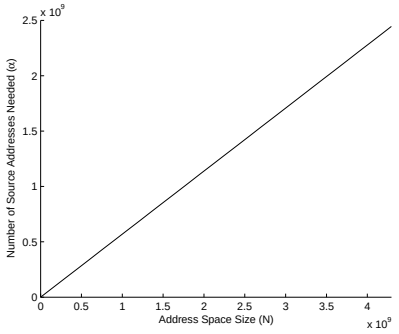

Fig. 1. Naive random scan against TRW ($P_s = 0.3$)

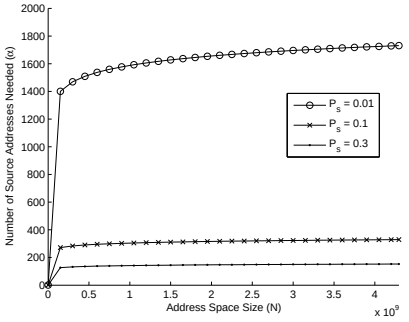

Fig. 2. z-Scan against TRW

4 Hybrid Detection Method and Evaluation

Section 3 has shown the vulnerability of Positive-Reward-based detection methods to distributed scans, where the attackers collude to create extra rewards for each other.

The other main type of scan detection methods shown in Section 2 are Timeout-based methods. It is well known that Timeout-based methods are easily eluded by using delayed probing, i.e., sending probing packets with enough time delay between them to allow expiration of previous events, so the anomaly score does not increase.

Positive-Reward-based methods based on successful connections, such as TRW, are resilient against evasion attacks using delayed probing. For example Weaver et al [22] show a TRW variant that can detect attackers probing at rate larger than one probe per minute. On the other hand, Timeout-based methods will not be eluded by a z-Scan. We

propose then to combine both approaches to create a scan detection method which is highly resistant to known evasion techniques. We call it a hybrid detection method.

In the remainder of this section, we present a simple example of the hybrid detection methods using TRW and Block Scan Detection (BSD) and show how the detection methods can complement each other. After a brief analysis on delayed probing against BSD, we provide numerical analysis on the robustness of the hybrid detection method.

4.1 Delayed Scan against BSD

BSD methods usually work as follows. There are two parameters: a time window of fixed length w and a threshold value t. BSD keeps a counter for the current number of events for each remote IP address. Examples of usual events are the number of destinations contacted or the number of unsuccessful connections sent. Every time a new event occurs, the counter is incremented and compared to the threshold. If the threshold has been exceeded an alarm is thrown. Each event has an age of length equal to the value of the time window parameter. The age of an event is set to zero when the event is observed, and after the age has become larger than the time window, the event is expired and the counter is decremented.

Timeout-based methods are easily eluded by using delayed probing. In particular, BSD methods can be eluded by sending probing packets with enough time delay between them to escape a preset time window and never reach the preset threshold.

When the attacker can determine the values of the window and threshold parameters and is free of time limit, it is able to scan the whole address space without being detected and using a single IP address. However, when a constraint is given on time T, the attacker should probe simultaneously using multiple source IP addresses to evade the detection of BSD.

Since α source hosts should complete probing N target addresses in the protected network address space:

$$\alpha r T \geq N$$

where r is a probing rate (the number of probes per time tick). In addition, the number of events per time tick should not exceed $\frac{t}{w}$, where w is the size of the time window, and t is the threshold for the number of events allowed in that window. Assuming the BSD method uses failed connection attempts as events (e.g. Kato et al), then the maximum probing rate r_{max} is:

$$(1 - P_s) r_{max} = \frac{t}{w}$$

If the BSD method uses any probe attempt as event (e.g. Snort) then $P_s = 0$ in the above expression.

By setting r in Equation 10 to r_{max}:

$$\alpha \geq \frac{1 - P_s}{T} \frac{w}{t} N \tag{10}$$

This result shows that the minimum number of addresses that an attacker, using a delayed scan against BSD, needs to complete the scan, when subject to a time constraint T, grows linearly with N.

Figure 4 plots α as function of the size of the address space of the scanned network N for different values of T. The window, threshold and fraction of active hosts are set to: $P_s = 0.3$, $t = 10$ probes, $w = 600$ time ticks. As shown, delayed scan evades BSD with only one source IP address when it has sufficient time ($T = 180,000$ time ticks); but otherwise α is directly proportional to address space size N.

4.2 Hybrid Detection Method

In our hybrid detection method we adopt a combination of TRW and BSD. We show that the hybrid detection method forces the attacker to use more addresses in order to complete the scan when compared with the case where only one of the two methods is deployed. In this hybrid method, we assume that TRW and BSD are operating independently in parallel, and a detected scan source will be blocked regardless of which detection method detected it. Simply put, the attackers' scan efficiency is bounded by the more effective of the two detection methods against the attackers' strategy.

In the remainder of this section we show how the Hybrid detection method performs when faced with three different scan techniques: z-Scan, delayed scan and a combination of both.

z-Scan against Hybrid detection When z-Scan is performed against BSD, assuming that the time needed to send probing packets is relatively small compared to the time window of BSD, we can suppose that all failed connection attempts will fall within the window. Therefore, the number of times the attacker will be blocked is $\alpha = \frac{N(1-P_s)}{t}$, where t is the threshold of failed connection attempts within the time window of the BSD methods.

Delayed scan against Hybrid detection Conversely, if delayed scan is performed against TRW, α is equal to that of performing a naive random scan against TRW as shown in Section 3; TRW is, theoretically, irrelevant to the time frame where the probing events occur. Thus, from Equation 4:

$$\alpha_{trw} \geq N \frac{P_s \log \frac{\theta_1}{\theta_0} + (1 - P_s) \log \frac{1-\theta_1}{1-\theta_0}}{\log \eta_1}$$

Since the Hybrid detection method uses both TRW and BSD in parallel, then:

$$\alpha = min\left(\alpha_{trw}, \alpha_{bsd}\right) = min\left(N\frac{P_s \log\frac{\theta_1}{\theta_0} + (1-P_s)\log\frac{1-\theta_1}{1-\theta_0}}{\log\eta_1}, \frac{1-P_s}{T}\frac{w}{t}N\right)$$

Accordingly, the values of α in both z-Scan and delayed scan are linearly bounded by TRW and BSD. As Figures 3 and 4 show, the hybrid detection method forces both z-Scan and delayed scan to use a number of addresses that is linear with the size of the address space. However, these results are dependent on multiple parameters for configuring detection methods and network environment such as t, w, T, and P_s.

Combined scan against Hybrid detection The attackers can also combine the evasion techniques to elude this hybrid detection method. We give a simple example of combining delayed scan and z-Scan to evade the hybrid method provided in this paper. That is, the attacker can perform z-Scan with a low scan rate set to evade the threshold value of BSD. In order to complete the scan of the address space of size N in time T, the attacker needs to divide the address space into different *subspaces* and simultaneously perform z-Scan on each subspace using multiple addresses. If the address space is divided into D subspaces of the same size, each subspace is of size $\frac{N}{D}$, and the active host ratio in that subspace is P_s. Then, the number of source addresses, σ, needed to scan one subspace using z-Scan is, from Equation 9:

$$\sigma = \log_k\left(1 + \frac{P_s(\frac{N}{D} - n_1)}{s_1 + \frac{l}{k-1}}\right) + 1 \tag{11}$$

To evade BSD with the threshold value of t and the window size of w, the maximum scan rate r_i in the i-th round of z-Scan should satisfy:

$$(1 - \frac{s_i}{n_i})r_i = \frac{t}{w}$$

In addition, since each z-Scan task on a subspace should be finished within the constraint T, the sum of time consumed in each round of z-Scan should be equal to or less than T.

$$\sum_{i=1}^{\sigma}\frac{n_i}{r_i} \leq T \tag{12}$$

Finally, the total number of source addresses, α, needed in this combined scan is $D\sigma$. Since there can be multiple possible values of D which meet constraint T, the attacker can choose the minimum of the possible $D\sigma$ values which satisfy equation 11 and 12. Therefore:

$$\alpha = min\{D\sigma \mid \sigma \text{ and } D \text{ satisfy Equation 11 and 12}\} \quad (13)$$

Through numerical iterations, we obtain the values of α with respect to N. The results indicate that the number of source addresses the attacker needs in the combined scan is proportional to the address space size N. So, as expected, when faced with a hybrid detection method, the attacker would prefer the combined scan, since it achieves better performance than the individual z-Scan or delayed scan methods.

Figure 5 shows how the combined scan performs in the face of a hybrid detection method with varying values for the time constraint T. Clearly, the more constrained the attacker is (i.e. smaller values of T) the larger the number of addresses it needs to use to complete the scan in the given time.

Limitations Even though the hybrid approach provides higher effectiveness in detecting evasion attacks, it has several limitations. In terms of administrative efficiency, it requires additional cost for deploying and integrating multiple scan detection systems. Regarding the detection efficiency, simply combining detection results could aggravate the total false positive rate; *i.e.*, the false positive rate of the hybrid method is additive.

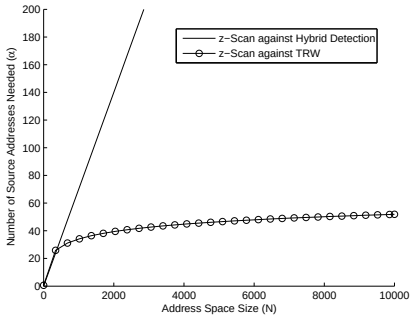

Fig. 3. z-Scan against Hybrid detection and TRW ($P_s = 0.3$)

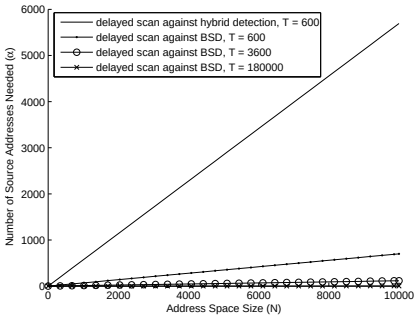

Fig. 4. Delayed scan against Hybrid detection ($P_s = 0.3, t = 10, w = 600$time ticks)

5 Information-Hiding Countermeasures against Evasion Techniques

In Section 4 we presented an hybrid detection method to thwart both z-Scan and delayed scan. The presented hybrid method tries to thwart the attacker's evasion attack. Another promising type of defense is rather than raising the bar for the scan technique, try to hide the topology of the network, thus reducing the utility of the scan itself. In this

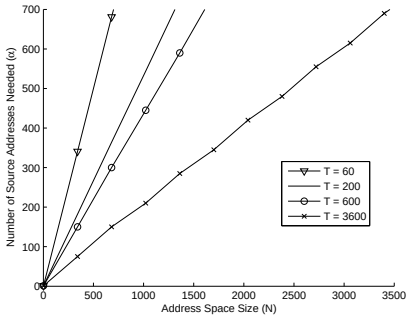

Fig. 5. Combined scan against hybrid detection ($P_s = 0.3, t = 10, w = 600$ time ticks)

section, we explore one such countermeasure. Despite its simplicity, we show that the effectiveness is promising in augmenting the current detection methods to curb evasion attacks.

The methods we study is *All-Positive Response* (APR). APR is a technique that gives false responses when receiving packets destined to unassigned IP addresses or to closed ports on active hosts. The generated responses falsely indicate that a host exists on that address and has the probed port open. From the attacker's point of view, information obtained during the scan cannot distinguish which host is active and which port is open since all of them appear active/open. In addition, the APR method can be easily implemented by applying virtual honeypot technology [16].

There are other such countermeasures that could potentially help against evasion using z-Scan, such as Antonatos et al.'s Network Address Space Randomization (NASR) [7], where hosts are forced to periodically change their IP addresses. We leave the study of such other countermeasures for future work.

z-Scan against TRW with APR Since the z-Scan technique is highly dependent on the set of known active hosts, say s_a, we can show that its performance against APR will be significantly degraded. When TRW is employed with APR, contrary to the attacker's expectation, s_a is just a set of addresses with active hosts ratio P_s. Without APR, all hosts in s_a would be active.

Therefore, in this case, the z-Scan behaves similar to a naive scan. Initially, when an attacker begins z-Scan, it will be blocked after n probes as shown in Equation 3. At that point the attacker believes that it knows a set of n active hosts when in fact only nP_s of those are active.

Thus in the next round when the attacker alternates one probe to an unknown address with one probe to an address it believes to be an active host, TRW will detect it and block it after n probes because the fraction of active hosts in the probes will be P_s, the same as in the case of a naive scan. So, in every round each scanner address from the attacker is allowed to probe n addresses rather than an increasing number with z-Scan.

Even worse for the attacker, in this case only half of n is newly-scanned hosts since the attacker alternates probing targets between the addresses in s_a and a randomly selected probing target (from the rest of the address space). This cycle will be repeated until the whole address space is scanned. To summarize, the attacker can probe n new hosts in the first round and $\frac{n}{2}$ new hosts each in the subsequent rounds. Thus, the number of source addresses required for the attacker is:

$$\alpha = \frac{N - n}{\frac{n}{2}} + 1$$

that is:

$$\alpha = \frac{2N}{n} - 1 \tag{14}$$

which is about twice as large as that of naive scan against TRW.

Thus, z-Scan is completely inefficient against APR since a naive scan would approximately require half the number of addresses. In this case the use of APR in the protected network forces the attacker to use a more sophisticated probing technique. A nice property of information-hiding countermeasures is that they can be combined with any scan detection method such as the proposed hybrid detection method, to form a more complete defense solution that both obscures the topology of the network and raises the bar for the scan techniques used by the attacker.

6 Related Work

There has been a wealth of research on scan detection methods. Early proposals such as the Network Security Monitor (NSM) and the old Snort scan detection method (portscan preprocessor) [19, 11] counted probes in a fixed window of time, flagging a external host as a scanner if the probe count exceeded a preset threshold.

Following work built on the observation that unsuccessful connections are a better indication of scanning than just the number of probes generated by a host [15, 18]. The performance of these methods greatly varied with the values of its parameters.

More recent work also using unsuccessful connections as events, employs a random walk framework to decide between the hypothesis that a remote host is a scanner or benign [12]. Followup work using the random walk framework includes [20] where the authors focus on detecting internal, rather than remote, scanners present in the monitored network. It also includes [22] where the authors use several approximations in order to limit to a minimum the resources (e.g. memory) needed to operate it.

There is a separate group of scan detection methods that assigns anomaly scores to events, based either on the access probability for each internal host [14] or conditional probabilities extracted from the addresses and ports pairs [21].

There has been little previous research on evasion techniques. Ptacek and Newsham show different insertion and evasion techniques that affect Intrusion Detection Systems [17]. There is also previous research work on overloading IDSs [9] and several tools have been developed with the same purpose [5, 6]. Some tools have been created for information-hiding at the end host, such as Morph which allows the user to emulate any operating system by forging replies to probes [4]. In general, most evasion work comes from the underground literature [1, 3].

7 Conclusion

Numerous approaches have been proposed to detect network scans. However, despite the importance of limiting the information obtained by the attacker, and the wide availability of such scan detection methods, there has been very little research on the evasive scan techniques, which can potentially be used by attackers to avoid detection. In this paper, our contributions are five-fold.

First, we categorize current scan detection methods using a novel point of view, their amnesty policy. Such a classification allows us to distill the essence of each class of detection methods and facilitate us in analyzing their vulnerability to evasive scan techniques and countermeasures. Second, we propose two novel metrics to measure the resources that an attacker needs to complete a scan without being detected: the time and the number of IP addresses; needed by an attacker to complete the scan of a certain network space, while remaining undetected.

Third, as a concrete example demonstrating evasive scans against Positive-Reward based detection methods, we propose a new distributed evasive scan attack, z-Scan, which is extremely effective against TRW. With z-Scan, an attacker can completely the scan of a given IP address space using only a small number of different IP addresses to send scans from (where the number is only logarithmic to the size of the IP address space to be scanned).

Fourth, as a countermeasure, we propose a hybrid approach which combines Positive-Reward and Timeout-based methods and demonstrate its effectiveness against evasive scans through analysis and simulation. Finally, we also study information-hiding countermeasures, where we actively respond to scans with false information, and demonstrate that this type of countermeasures are extremely effective against evasive scan attacks. Moreover, the hybrid approach and the information-hiding based countermeasures are complementary, and can be combined for even greater benefits. Approaches such as hybrid detection and information-hiding based countermeasures have not been well studied before. We hope this work will serve as a first step and encourage more study in this direction.

References

1. antirez. IP ID reverse scan. December 18, 1998. http://www.kyuzz.org/antirez/papers/dumbscan.html.
2. Fyodor. The Art of Port Scanning. Phrack 51, volume 7. September 1, 1997. http://www.phrack.com/phrack/51/P51-11.

3. hybrid. Distributed information gathering. Phrack 51, volume 9. September 9, 1999. http://www.phrack.org/phrack/55/P55-09.
4. Morph. http://www.synacklabs.net/projects/morph/.
5. Snot. http://www.l0t3k.org/security/tools/ids/.
6. Stick. http://www.l0t3k.org/security/tools/ids/.
7. S. Antonatos, P. Akritidis, E. Markatos, and K. G. Anagnostakis. Defending against Hitlist Worms using Network Address Space Randomization. *ACM Workshop on Rapid Malcode (WORM 2005)* (Fairfax, VA, USA, 11 November 2005).
8. R. Basu, R. K. Cunningham, and R. P. Lippmann. Detecting Low-Profile Probes and Novel Denial-of-Service Attacks. *Proceedings 2nd Annual IEEE Systems, Man, and Cybernetics Information Assurance Workshop* (West Point, NY, USA, 5–6 June 2001).
9. S. Crosby and D. Wallach. Denial of Service via Algorithmic Complexity Attacks. *Proceedings of the 12th USENIX Security Symposium (USENIX 2003)* (Washington DC, USA, 4–8 August 2003).
10. H. Dreger, A. Feldmann, V. Paxson, and R. Sommer. Operational Experiences with HighVolume Network Intrusion Detection. *11th ACM Conference on Computer and Communications Security (CCS 2004)* (Washington DC, USA, 25–29 October 2004).
11. L. T. Heberlein, G. V. Dias, K. N. Levitt, B. Mukherjee, J. Wood, and D. Wolber. A network security monitor. *Proceedings of the IEEE Symposium on Research in Security and Privacy*.
12. J. Jung, V. Paxson, A. W. Berger, and H. Balakrishnan. Fast Portscan Detection Using Sequential Hypothesis Testing. *2004 IEEE Symposium on Security and Privacy* (Berkeley/Oakland, CA, USA, 9–12 May 2004).
13. N. Kato, H. Nitou, K. Ohta, G. Mansfield, and Y. Nemoto. Real-time intrusion detection system(IDS) for large scale networks and its evalutations. *Proceedings of IEICE Transactions on Communications*.
14. C. Leckie and R. Kotagiri. A Probabilistic Approach to Detecting Network Scans. *Proceedings of the Eighth IEEE Network Operations and Management Symposium (NOMS 2002)* (Florence, Italy, 15–19 April 2002).
15. V. Paxson. Bro: a system for detecting network intruders in real-time. *Computer Networks (Amsterdam, Netherlands: 1999)*, **31**(23–24):2435–2463.
16. N. Provos. A Virtual Honeypot Framework. *Proceedings of the 13th USENIX Security Symposium (USENIX 2004)* (San Diego, CA, USA, 9–13 August 2004).
17. Thomas H. Ptacek and Timothy N. Newsham. *Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection*. Technical report.
18. S. Robertson, E. V. Siegel, M. Miller, and S. J. Stolfo. Surveillance Detection in High Bandwidth Environments. *Proceedings of the 2003 DARPA DISCEX III Conference* (Washington DC, USA, 22–24 April 2003).
19. M. Roesch. Snort-Lightweight Intrusion Detection for Networks. *Proceedings of LISA'99: 13th Systems Administration Conference* (Seattle, WA, USA, 7–12 November 1999).
20. S. E. Schechter, J. Jung, and A. W. Berger. Fast Detection of Scanning Worm Infections. *7th International Symposium on Recent Advances in Intrusion Detection (RAID 2004)* (Sophia Antipolis, French Riviera, France, 15–17 September 2004).
21. S. Staniford, J. A. Hoagland, and J. M. McAlerney. Practical Automated Detection of Stealthy Portscans. *Proceedings of the 7th ACM Conference on Computer and Communications Security (CCS 2000)* (Athens, Greece, 1–4 November 2000).
22. N. Weaver, S. Staniford, and V. Paxson. Very Fast Containment of Scanning Worms. *13th USENIX Security Symposium (USENIX 2004)* (San Diego, CA, USA, 9–13 August 2004).