A Context Aware Scan Detection System

Joel Scanlan and Jacky Hartnett,

University of Tasmania, Hobart, Australia

Summary

It is well known that intrusion detection systems can make smarter decisions if the context of the traffic being observed is known. This paper examines whether an attack detection system, looking at traffic as it arrives at gateways or firewalls, can make smarter decisions if the context of attack patterns across a class of IP addresses is known. A system that detects and forestalls the continuation of both fast attacks and slow attacks across several IP addresses is described and the development of heuristics both to ban activity from hostile IP addresses and then lift these bans is illustrated. The system not only facilitates detection of methodical multiple gateway attacks, but also acts to defeat the attack before penetration can occur.

Key words:

Intrusion Detection, .Scan Correlation, Prevention

1. Introduction

During the last decade the numbers of networked computers globally has increased astronomically due to the rise of the internet. As a consequence the threat of attack against computer systems is very real, resulting in network security becoming one of the most important priorities not only for system administrators, but for the average network user.

Just as virtually every email user has received spam and virus emails, effectively every computer has had its ports probed, been infected by a virus, or been trivially (or extensively) attacked by another user. While the internet does connect us to the “Information Super Highway” it also allows malicious users to use the same highway to attack any other user or server connected to it either, directly or through shared network access.

It is at this point that network security infrastructure steps in to provide protection to users from malicious users and their attacks. There are two main types of protective infrastructure which is widely deployed on networks currently: Firewalls and Intrusion Detection Systems. Firewalls act as a means of access control, allowing, and disallowing, access into and out of a given network; Intrusion Detection Systems are designed to detect any malicious behaviour which is occurring on the trusted side of a Firewall

Many networks have grown so large that they often possess several access points to the Internet (or to other networks), each protected by an individual firewall or gateway. Intrusion detection systems (IDS) situated on each of these gateways try to work out when the packets that arrive have potential to harm the network that they are protecting and take defensive measures; usually by dropping the packet. Profiles of attacks are built up until the firewall is sufficiently convinced that all packets from a particular source should be dropped, at least for the time being. Each IDS usually works in isolation and independently comes to its own conclusions. This means that an attack targeted at a particular organisation can be disguised by spreading the indications of the attack across several gateways such that no one gateway can be sure that it is detecting an attack. It also means that for automated attacks that work their way through a range of network addresses, gateways that have detected an attack, especially one previously unseen, and allowed some damaging packets into the network have no way of communicating the ‘watch out for this’ knowledge that they have accumulated.

This paper explores the benefits that could be gained should a group of gateways be able to communicate the knowledge that each has independently gained by investigating the output from the logs of a group of related gateways (similar to that in Figure 1). The work examines whether or not it is possible to detect that an attacker is interested in multiple gateways not just this gateway. Finally, we describe using live logs files to determine empirically an optimum length of time for which to keep banning firewalls rules in place.

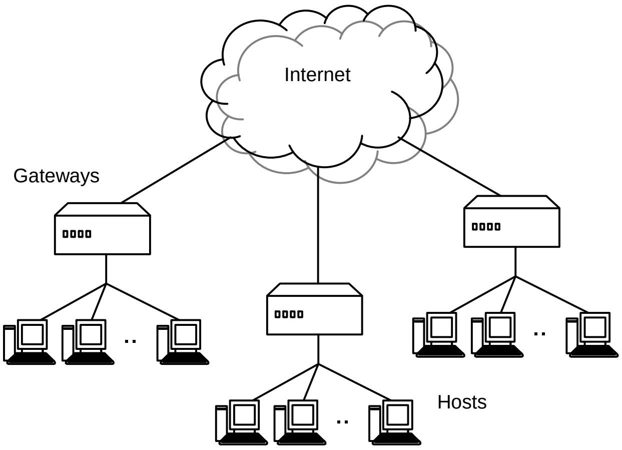


Figure 1. The work in this paper examines a small group of gateways arranged in a similar topology to those in this diagram.

Manuscript received January 5, 2008
Manuscript revised January 20, 2008

2. Log Analysis

A fundamental component of any system which monitors network activity is the Audit Log. The audit log records all of the activities which have happened at a given place within the network; whether it is a gateway or another type of network monitoring sensor sniffing packets off the network directly for analysis.

Attacks are generally identified by Intrusion Detection Systems examining audit logs by one of the following two methods: Anomaly Detection and Signature Detection.

The Anomaly detection method requires a profile of each user or user group to be made, this is usually learnt by monitoring their normal behaviour [1, 2]. The behaviour model is then compared to user actions upon the system, searching for behaviour that does not fit the model; this behaviour is then classed as abnormal behaviour and treated as an intrusion.

Anomaly detection is broader than just mapping profiles of human usage. It is also applicable to processes and network access or usage [3]. Network traffic analysis can yield profiles of normal usage that can be used in monitoring network traffic for anomalies and thus to detect attacks.

The Misuse detection method searches audit logs for known attacks, matching malicious behaviour to pre-defined signatures. Misuse detection has a database of attack signatures against which it can compare network event patterns in order to discover an attack. This results in signature detection systems being able to be operational directly after they are installed without the need for any training of the system [2].

Misuse based intrusion detection is significantly more computationally efficient than anomaly based detection per item of knowledge as it does not need to create matrices for each system activity [4], in order to compare it to a users normal activity (i.e. to decide on whether activity is enough of an anomaly to warrant a detection alert). Misuse detection has a flaw in that it requires a signature for a given attack to be able to be detected, and in some instances this is a case of waiting for an attack to occur, to then be able to make a signature to protect against it.

3. Detection Context

The context of network traffic as a topic in network security has grown in importance over the last decade. Initially the focus was only on Firewalls, which have progressed from being simple packet filters to more context aware proxy and dynamic packet filters [5]. Likewise, various Intrusion Detection Systems have been designed and modified to also utilize and act upon information based on the wider context of an environment, or users' behaviour [6-8]. Further, researchers have also amalgamated alert data from multiple detection sources targeting services such as web and mail servers [9].

There are two primary reasons for the increased usage of contextual information by these two types of network infrastructure: efficiency and accuracy. The initial usage of contextual network information was to decrease the number of packets examined by a firewall; by ignoring packets in an already authorised session. Conversely, Intrusion Detection Systems have tended to use contextual information in conjunction with attack signatures to more accurately detect an attack.

Current Intrusion Detection systems using signature or anomaly detection (or a combination of both) are able to operate effectively on a single gateway or as a network sensor. They analyse network traffic within the context of a single gateway; however if a user attacks multiple gateways upon the same network they are usually treated as a single gateway attack at each gateway, meanwhile any breech which does occur is effectively a breech upon them all. A scan which may appear to be trivial at each gateway is in the context of the entire network actually a coordinated attack, and of far greater interest to the network as a whole.

Detection systems attempting to discover scans which target more than a single gateway on a network need to have a complete knowledge of the given network's context. This entails the system having access to the audit log at each gateway upon the network, and not merely a listing of alerts or threats from detection software which are examining behaviour in a single gateway context. The result is a secondary stage of analysis of network events.

4. Port Scans

Throughout much of the research and development of intrusion detection systems scans of hosts and ports have been described as the reconnaissance portion of an intrusion [10], and only shown to be of use once further malicious activity has occurred. Due to the sheer number of scans that occur, it has been far too computationally expensive to attempt to correlate scan data. As only a small percentage of scans ever translate into full blown hacking attempts this has not been seen as a serious concern. However, recent research is tackling the problem of correlating this data in an efficient manner. If the data can be analysed and correlated, then further attacks may be thwarted, or scan profiles from repeat offenders developed. The precursor nature of reconnaissance activity can then be used as a defensive mechanism.

There has been a large amount of research on the analysis of audit logs and network activity within the context of intrusion detection and much of the knowledge learned can be translated to scan correlation. Many challenges have been overcome in the past 20 years enabling ID systems to scale to large networks while at the same time remaining effective. Scan correlation now hopes to achieve these same twin goals. Before we examine the existing scan correlation systems, it is appropriate to take a deeper look at why scans occur.

Every IP address gets scanned. There are a finite number of IP addresses; it is the way that network addressing was designed. To re-visit the often used analogy of a hacker being like a burglar breaking into a computer instead of a home. Imagine a burglar who has access to an address book of every house in the world, and can find out a few details such as whether anyone lives there and what the alarm system is probably running without leaving the relative safety of his own home. This is why every computer gets scanned. Hackers have an address book of possible locations, and with a handful of scans that last only a few seconds they can discover whether a computer is at the address and what services it is running.

There are many readily available tools which allow for various automated scans to be completed at the click of a button. The most commonly used probing utility which is used both by system administrators and malicious users alike is called nmap [11]. Nmap allows for a wide range of different scans to be completed over various IP ranges or lengths of time. It comes equipped with over of 700 different recognisable operating system finger prints to match to scan results. Furthermore, it also allows for various scans that guarantee anonymity such as idle scans. An idle scan involves using a second computer as an intermediary to hide the identity of the scanner [12]. The attacking PC probes the target, spoofing the source address as being from an idle computer, while at the same time constantly probing the idle PC. If the idle computer is only responding to the hacker's activities, the ID in the header of the response packets coming back to the hacker will increment when the target PC responds to the spoofed packet. This allows the malicious user to know that the target IP and port are indeed present and open.

Reconnaissance activity occurs in every field where there is an attack that is about to take place. Likewise, malicious computer users also probe systems ahead of a more direct attack, looking for targets and then weaknesses. In this way, scans are undertaken across vast amounts of IP space, first mapping the locations of gateways, networks and hosts, before then probing these computers looking for vulnerabilities to attack. However there are also scan activity which is benign in nature, originating from sources such as web crawlers and proxies. Often these types of services appear as a scanner, but are totally benign. One of the challenges facing scan correlation systems is to classify both benign scan activities as well as malicious.

5. Scan Correlation Systems

This section will examine two existing scan correlation systems that have been produced to date. The first of which is the most similar to the system that we produced, while the second is an interesting system which is highly relevant in the further work discussed later in the paper. The system that is discussed in this paper and the system discussed in section 4.1.1 both operate using a statistical anomaly based method involving thresholds. Existing IDS systems such as Bro [13] and Snort [14] both make use of thresholds in the context of scans. Although both systems use static values, one of 20 and one of 100, with no apparent justification for the value chosen. The two systems discussed here that use thresholds were projects that focused on finding the appropriate threshold level to efficiently classify scanners.

5.1 Threshold Random Walk: Sequential Hypothesis Testing

At the same time as this work was being undertaken other similar work was being completed and published by Jung et al (2004). This system focuses primarily on distinguishing between the benign scans that take place and the malicious scans. The question that Jung et al were answering was how to detect when a scanner is malicious and when it is benign. Both our work and Jung's, at a fundamental level, involve simple profiling of scan behaviour, to extract meaning from otherwise noise-laden scan activity.

Jung et al [15] proposed a detection algorithm called Threshold Random Walk (TRW). The algorithm is based on the mathematical technique called Sequential Hypotheses Testing described by Wald [16]. The basis for the algorithm is that scans or failed connections to unpopulated IP space are much more likely to come from a malicious user than an authorized user. As a possible scan takes place for each host that is probed the TRW algorithm notes whether the source IP was successful or not, with special attention to whether or not the destination IP was in use or not. For the source IP a tally is kept of the results, and once that value reaches a threshold level a decision is made as to whether the source IP was an unauthorised scanner or not. The method has given good results, out performing both the Bro and Snort intrusion detection systems when detecting scanners with a threshold of 4 being found as the optimum. Web crawlers and proxies were able to be distinguished from regular scanners because they rarely probed unpopulated space.

While the system gave good results on the networks on which it was tested, it would be interesting to see how it would perform on a more heavily populated IP space. The best results occurred on a network which was only 4.47%

populated, and with the second network tested being only 42% populated. The system seems to have been built overlooking how many companies, government departments and countries are running out of IP space, or are at least avoiding attaining more, using the bulk of the IP space they own.

5.2 Spade and Spice: Simulated Annealing

The most ambitious project that has been proposed to date in scan correlation is the Spice and Spade System by Stainford et al (2002). The system was being built with Defense Advanced Research Projects Agency (DARPA) funding in the U.S. However, in the fallout from 9/11, all non-classified research projects were either brought within the Department of Defense, or lost their funding. The latter happened to this project before it was completed. The authors have now moved on to other projects.

The work that had been proposed, and partially implemented, by Stainford et al [17] was intended to correlate scan activity and would not only classify the user as a scanner, but also be able to link it to past scan activity. The goal of such a connection would be to link users who operate over a long time under multiple source IP addresses to avoid detection. This is precisely what is required in a system that identifies and tracks reconnaissance activity. The system they proposed operates with two components, a sensor (Spade) and a correlation engine (Spice).

The proposed system involved Spade feeding events into Spice, along with an anomaly score it had generated based on the source IPs activity (the negative log of the probability of the event occurring). Spice then places the event in a graph noting the various properties of the event such as source IP, target IP, target and source port and time. The location at which the item is placed into the graph is found through the use of a search algorithm called Simulated Annealing. Prior to each event being added, the graph is searched to locate the best location for it, placing it near events of a similar nature. It is here that past scans and events can then be correlated to determine whether a source IP has been changed. It could also be used to decide on the correct response to the user's activity. However for the reasons mentioned above, Spice was never implemented.

The approach appears to have a significant potential and its results would have been very interesting if it had been implemented. The paper which detailed the proposal mentioned how it could be of great use implemented in a distributed fashion, allowing for a more robust system that would scale to large networks. While this work was not completed it demonstrated a possible advance in the possible methods for scan correlation, and is discussed in the further work of this paper.

6. Implementation

The work discussed in this paper has been undertaken in two phases, each comprising a year of research as an honour topic. The first phase was aimed at testing the hypothesis that it was possible to detect scans which occur against multiple gateways upon a single network. The results of this phase will be briefly discussed in this paper, however they have been outlined further in Scanlan et al. [18]. The second phase of the research has recently been completed and this paper will discuss a selection of interesting results from this work.

6.1 Phase 1

The initial work carried out was designed to test the hypothesis that a malicious source IP address could be detected scanning multiple gateways upon the same network through centralised analysis. The system which was implemented analysed actual audit data from a gateway range that consisted of multiple remote gateways tied to a central server (ns1) which conducts the analysis, as shown in Figure 2. Ns1 is effectively bound to the IP addresses of almost a complete C-class running from 0 to 252 in the last octet, through monitoring the 5 gateways that divide the IP space. The resulting IP range of 'virtual' consecutive gateways, as it appears externally to be 253 separate machines when really each IP address will report to the same machine in one amalgamated log, through amalgamating the actual gateway logs. The audit log still reports which IP address within this range was probed, meaning that it is possible to analyse the data for a single IP address within the 253 range. The majority of the log data used in this study is comprised of simple port probes which have been targeted against IP's within the range.

As the goal of the system is to know what is happening across multiple gateways, and therefore the whole network, the retention of context is vitally important. The context of the amalgamated log is preserved through the way in which the log is parsed by the system. The system contains 2 modules: the Analysis module and the Tracking module.

The Analysis module stores a simple profile of each IP which probes the network in a database. This profile consists of such information as IP, target gateway, target port, date and time, ID number, two Boolean values and a probe count. The two Boolean values are used to store whether or not the given user has probed more than one IP address or more than one port. This first Boolean value is of much greater importance then the second, as it plays a crucial role in the second goal. This Audit table effectively compresses the amalgamated log down to being a single entry per malicious IP, indicating how often they have probed the system and whether or not their probes have been against multiple ports or gateways.

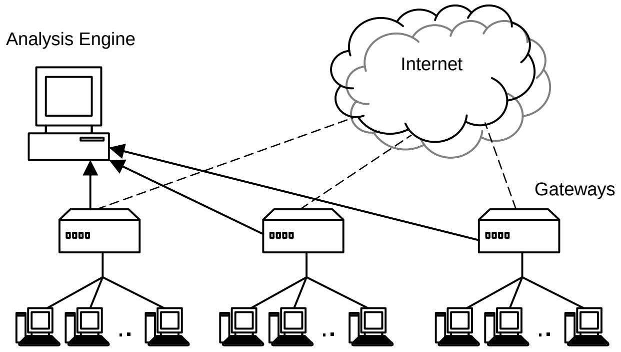


Figure 2. The activity log from each of the gateways is amalgamated into a single log for analysis.

The Tracking module examines each IP address at a much greater resolution, in terms of information being recorded, than the Analysis module. The tracking module is not used for every IP, but only those which appear to be of interest to the Analysis module through probing multiple gateways or ports. For each probe which is sent by an IP a new entry is added to the tracking table, recording the IP, target gateway (for each gateway and not merely the first gateway as in the Analysis module), target port, date and time. This allows for the activities of an IP to be closely examined in terms of attempting to define a scan pattern that could be useful for linking different attacking IP's with similar patterns to being the same user who is changing IP addresses.

The second goal of the system was to examine if a threshold level could be established and at which point this should be set. Then if the number of probes from a single user exceed the threshold, a classification as to whether the user is a threat or not can occur. This threshold works primarily upon the probe count and multiple destination IP Boolean value.

6.2 Phase 1 Results

The two main results from the first phase of the research, applied to our project goals: proving the hypothesis that attacks across multiple gateways could be detected, and deciding whether or not an effective threshold level could be established.

As described in section 4, the Analysis module records a series of details about each IP address that probes the network. This includes a count of their actual probes, and whether or not a source IP has probed more than a single gateway. Table 1 displays several statistics which were gathered by the Analysis module. The first section of these was gathered on the Phase 1 log file. This log file was about 10MB in size and covered the 10 day study period of the 1st of September 2003 till the 10th of September. During the Study period 6766 individual source IP addresses probed the gateways, of which 776 (11.5%) probed multiple IP addresses. This represents a sizeable risk to networks which had previously gone unnoticed; however noticing that a multiple gateway scan has indeed occurred is only the first step; being able to detect it efficiently is the second half, and the more valued challenge of phase 1.

Table 1. Individual Source IP Address

	Single Gateways	Multiple
Phase 1 log		
Source IP Addresses	5990	776
% of Total	88.53%	11.47%
Phase 2 log		
Source IP Addresses	67029	8948
% of Total	88.2%	11.85%

The analysis module records a count of the number of probes which have been sent by each source IP address against any of the gateways on the network. The second goal of the system was to see if this count could be effectively used to gauge whether or not a user is likely to probe multiple gateways. To investigate the usage of this simple heuristic several test levels were examined. When examining the count value it was found that 83% of source IP addresses sent 3 or less probes against the network. For this reason 3 was the first value we examined as a possible threshold value, followed by 6 and 9 as these were also values where a substantial drop off in probe counts were seen. This heuristic was then used with the Boolean signifying a multiple gateway scan in order to try and classify source IP addresses. The results showed that at a threshold level of 3 only 3.2% of IP's were classified as potentially performing scans on multiple gateways. With the 11.5% being perfect detection, this was a relatively poor result, detecting only 28% of target group. By comparison, when the threshold level was increased to the levels of 6 and 9, the results returned were 8.3% (65% of target group) and 9.7% (85% of target group) respectively. These results were much more acceptable, however not quite at the levels desired. Figure 3 illustrates the result from further testing of other threshold levels. The optimum level efficiency was found to be at an 11 probe threshold, detecting over 90% of the target group.

The threshold level signifies the point at which, if a user exceeds the level at a single gateway, they are highly unlikely to probe multiple gateways. The result is that it is possible to track and detect over 90% of users who attack multiple gateways upon a single network.

6.3 Phase 2

Phase 2, which is described in the remainder of this paper, builds directly on top of the system implemented to complete the work in Phase 1.

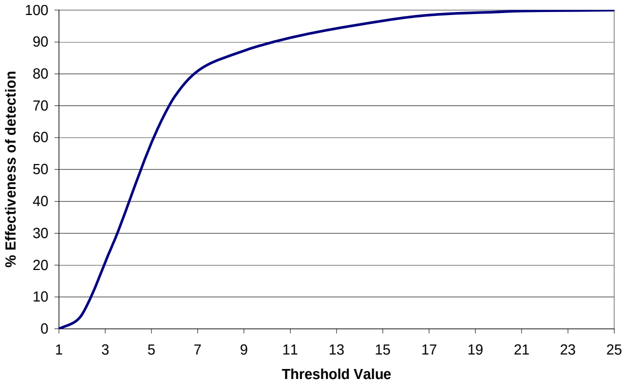


Figure 3. Effectiveness of Detecting IP address probing Multiple Gateways

The first goal in Phase 2 was to verify the work which was done in Phase 1 by using a larger, and longer (in terms of time) log file. Phase 1's log file was not a very large one and this validation is required to certify that the results can be duplicated over a longer time period.

However, once this is accomplished there are several other more practical goals to examine. One of these is the creation of an action module to respond to the users who scan multiple gateways upon the network. This module needs to protect the gateways on the network which have not been scanned by the malicious user ahead of their future attempted attacks.

The concept of adding additional rules to firewalls ahead of a network scan upon the given gateways when further scans may not result in attacks on each gateway is bound to have some administrators sceptical of the system. Concerns about the performance cost of the number of rules which are on firewalls are of great concern to system administrators, and have resulted in rule efficiency applications being produced [19]. Thus the idea of adding extra rules to a series of gateways needs to consider the possible cost in performance on those gateways, as it is obviously of greater concern than when a single rule is added to a single gateway. This is of increased importance in the context of adding rules in response to scan activity for large periods of time, and not more overt malicious activity. One aim of Phase 2 was to be able add and remove the rules it creates in real-time, with the length of the time which a rule is on a firewall to be optimised to be as short as possible, while still providing adequate protection.

For the removal of rules to occur in a timely fashion while still providing protection to the gateways of the network it was necessary to examine the results from the Phase 1 Tracking module to see in what way malicious users were scanning the network. Figure 4 illustrates the two main scans which were occurring during the Phase 1 log: fast scan and slow scan. The fast scans generally lasted just a few seconds, through to being a few minutes in total length. Figure 5 shows a classic fast example of a scan which lasts 1 min and scans all 250 IP's within the class C address. The slow scans last a far longer, often scanning at similar time intervals between probes lasting over several days. A third type of scan is a hybrid of the first two types, with a user doing a short fast scan, waiting 24 hours or longer, and then doing a second short fast scan. As a result of the differing scan types, an optimum ban time cannot be a static value and still be effective; it needs to be a dynamic value based on the activities of the given source IP. A further factor affecting the length of the optimum ban time is the length of time taken for users to respond to being banned and cease their attack. Figure 6 illustrates the how many probes users send after they have been banned from a gateway. 50% of users send 2 more probes after they have been blocked from a IP address, with 90% of users sent a further 11 probes or less before ceasing their activities at the given destination IP. The Figure 6 graph is based on the bans that were set in place (upon source IP's that only probed one address) during the period of time covered in audit log 1 described in Table 1.

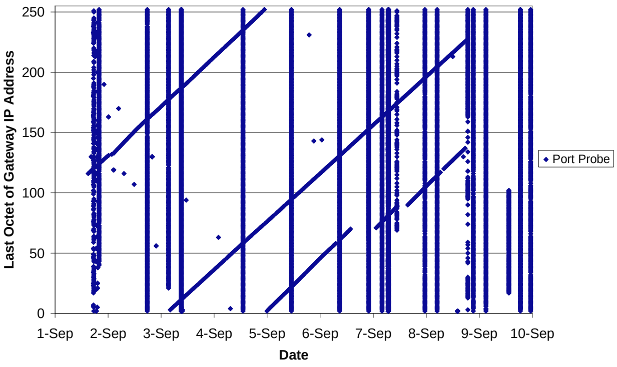


Figure 4. Ten days of gateway activity recorded by the Tracker Module

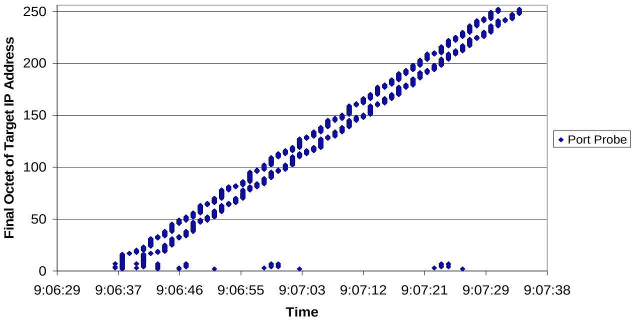


Figure 5. Scan across multiple gateways from a lone source IP address.

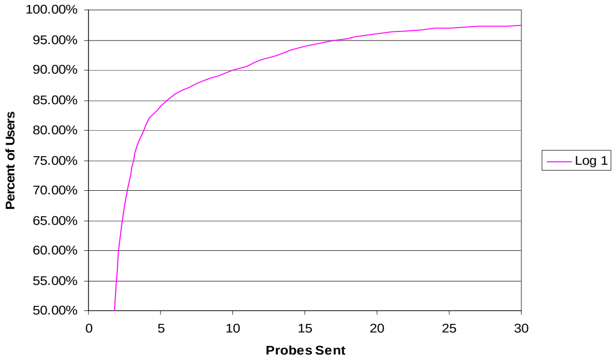


Figure 6. Probes sent from scanner after access has been blocked.

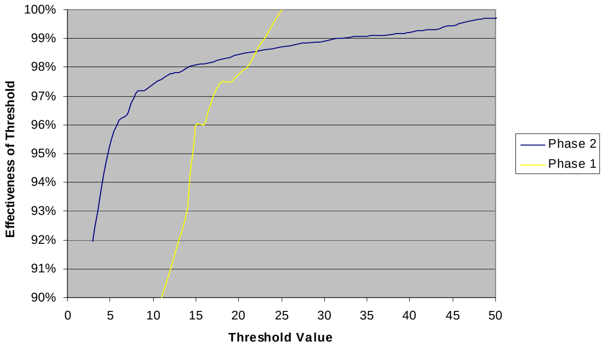


Figure 7. Comparison between Phase 2 and Phase 1 for optimum threshold level.

To be able to optimise the ban length without putting the system at risk from a malicious user the Action module also maintains a history of malicious IP's for a short time after their ban being lifted. This will result in the ability to reapply a ban to a repeat offender who returns after their ban has been lifted without them needing to go through the analysis process again, and risk breaking through the firewall and causing any damage to the internal resources of the network.

As the system is now moving more towards effective management of malicious scanners, and dealing with them efficiently. The Tracking module was disabled as its primary function is in depth profiling of a single users activity, to develop an effective threshold. This kind of profiling is picked up again in the further work by the authors.

6.4 Phase 2 Results

Phase 2's results can be split into two main groups: Phase 1 validation, and automated attack response.

The validation of the results which were produced by Phase 1 was the first priority before adding much further to the system. Optimisation tests were completed on the system, the results which are displayed in Table 1, with those of Phase 1 provided for comparison. The Phase 2 log is substantially larger than the Phase 1 log, being 270 MB in size, and covering the time period of July 1st through July 21st of 2004. The results showed that not only were multiple gateway attacks also detectable within a larger more comprehensive log; but were actually of similar frequency as discovered in Phase 1.

To fully validate the Phase 1 results, the optimum threshold level also needs to be calculated on the Phase 2 log. The Phase 2 log, similarly to the Phase 1 log, also had a large peak at 3 as a possible threshold level. A threshold level of 3 would result in a detection efficiency of above 90%; however that would result in ignoring quite a sizeable number of multiple gateway probing source IP addresses. Figure 7 clearly illustrates the increasing efficiency levels for different threshold values. The optimum, according to the Phase 2 log results, would indicate that a threshold around 10 would be the most efficient; this validates the threshold of 11 found by the Phase 1 log analysis. Figure 7 also plots the Phase 1 efficiency line for comparison; it clearly demonstrates the differences between the log files, but also their similarity in relation to threshold efficiency (both producing 90%+ for a value of 11, although a much smoother line is evident on the larger dataset).

The remainder of Phase 2 focuses on the way in which the system responds to a source IP which has been found to be probing multiple gateways. The initial task in this was to create an Action module for the system to use to deal with the discoveries made by the analysis module. The Action module adds and removes the source IP addresses from the Linux iptables based firewalls running on each of the network gateways. The module also maintains a network state within the database which allows for bans to be lifted upon their expiry.

The concerns mentioned in section 5.2 mean that the length of time which a rule remains in place needs to be as efficient as possible in order to keep the total number of rules to a minimum and preserve network performance. As a result the optimum ban time calculation will need to be both scalable and dynamic. The calculation will need to be scalable to enable it to work efficiently with short term bans in response to fast scans lasting several seconds; while still returning a larger ban time to enable the network to be protected from slow scans lasting several days. It is assumed that the vast difference between the different types of scans means a static value will be inefficient as it will result in being one of two things: far too long for fast scans in an attempt to provide protection for slow scans, or be too short in an attempt to be efficient for fast scans. This would only result in rules being reapplied multiple times for slow scans. To verify the value of a dynamic ban length we included a commonly used static value as a benchmark for comparison. The benchmark we used is that of a static 24 hour ban length which is commonly used by administrators to deal with malicious users.

For the ban length calculation to be well suited to each source IP address it needs to be individualised to the given IP. A straightforward way to do this is to record the mean time difference between each probe from the source IP addresses. The result is that the number returned for a scan lasting several days is far larger than the resulting number from a scan which lasted under 10 seconds. This mean time interval is the foundation of the ban length calculations we tested. The following were the initial calculations that were trialled:

> *Interval Squared: Squaring the interval allows for a ban length to be calculated based entirely off of the mean interval while still scaling quite high to provide protection to the network.*
> *Interval x Interval / 2: Similar to Interval Squared, however producing a shorter ban length in an effort to possibly attaining maximum efficiency.*
> *Interval x Threshold: While appearing to be chosen for convenience, this calculation is actually based off Figure 7 where the approximate optimum point for a interval multiplier is equal to our already existent threshold of 11.*
> *Static 24 Hour: This value is being used a pseudo benchmark as it is sometimes used by administrators.*

In addition to these calculations, a static value of 100 seconds is added to each result to allow for the cases where a source IP address probes extremely fast with multiple probes within a second (as in Figure 4). Without the added 100 seconds a mean time interval of zero seconds would result in a ban length of zero, thus resulting in the ban being lifted as soon as it is applied.

The initial results from the different ban length calculation methods which were trialled were on the disappointing side. The bulk of the methods returned results that were really not scalable as a ban length on a gateway. A sizable portion of several of the calculation produced acceptable ban lengths while the remainder tended to be far in excess of the real-world requirements. Upon examining the results it was found that the worst performers were those that generated the longest ban lengths, and with hindsight they were obvious poor choices for such a calculation. However the results did clearly show what was sorely missing from the dynamic calculations, but was inherent in the static value: a maximum ban length.

Despite producing ban lengths that were to long in some instances, the methods trialled were efficient in terms of the number of firewall rules which were added. Therefore to combat the problem, while endeavouring to maintain the existent efficiency another trial was run testing different values for a maximum ban length, initially using 3, 5 and 7 days. For the trial the calculation method of interval multiplied by the threshold value was chosen. This method was chosen as it was the most scalable of the calculation methods, which further highlighted the usefulness of user response time as shown in Figure 6 as being a valid metric.

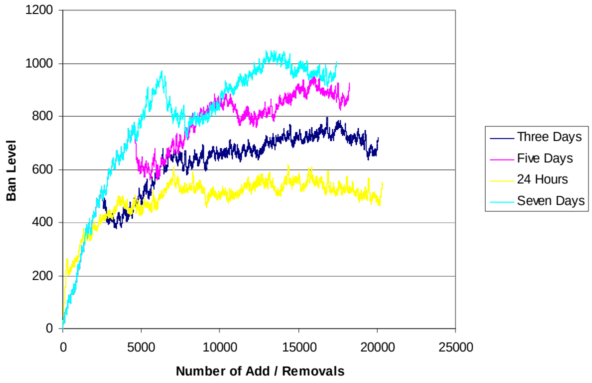


Figure 8. Comparison of Maximum Ban Lengths.

Figure 8 illustrates the effectiveness of the 3 different maximum ban lengths that were trialled. The smaller the ban lengths the closer the results got to the benchmark result from the static 24 hour value. However even using a 3 day maximum ban length the result was still markedly worse than that of the benchmark, while being a substantial improvement over the unbounded calculation methods. The calculation was still more efficient than the static value in the number of firewall rules added and removed from the gateways upon the network. However as the maximum ban length decreased this effect also decreased.

In order to try and maximise their two traits a 3rd trial was run which used a 24 hour maximum ban length. In conjunction with this an optimised version was also run which checked if a source IP had been active in the previous 2 mean time intervals. If the source IP has been active it would not lift the ban, but extend it by 50% of its original length as the attack was still ongoing. As a result, if a source IP's ban expires while it is still active, and would be reapplied once removed, it remains in place preventing an additional 2 firewall commands to remove it and then re-apply it.

The results shown in Figure 9 illustrate that the maximum ban length of 24 hours was more efficient then the static benchmark in terms of the number of firewall rules in place upon the network. However it did show an increased number of firewall rule commands sent across the network. In comparison the optimised version not only added and removed less rules then the static benchmark, but it also resulted in less rules in place then on the non-optimised system. The reason for this being that if a source IP is still active upon the network its second ban under the non-optimised version is twice as long. The source IP's which get banned twice remain banned for a shorter period of time under the optimised, lowering the overall total of IP's being blocked at the gateways. Therefore the most efficient method is the optimised interval multiplied by 11, with a maximum ban length of 24 hours.

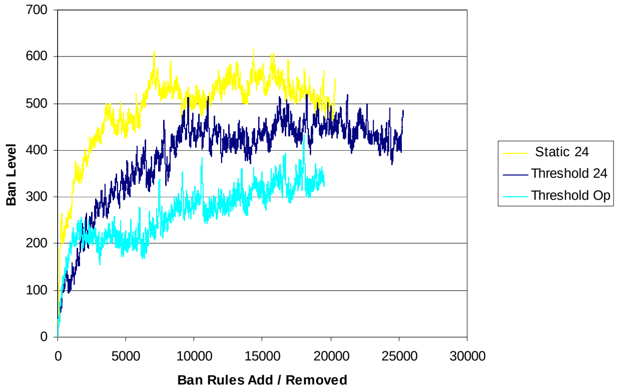


Figure 9 Comparison between the 24-hour Maximum Calculation and Static 24 hr.

6.5 Extra System Features

In addition to the system storing a state of the network in terms of current banned source IP's, the Action module also keeps a record for a short time of past banned users. This enables the system to have knowledge of a malicious IP without needing to have the IP blocked at the gateway. This longer term retention of contextual information allows for the analysis module to recognise an IP who has been banned previously to be re-banned by the action module without needing to again be analysed as being a threat. This 'safety net' allows for additional protection against attackers who do not fit the model of behaviour which suits most source IP addresses.

The Ban History, Banned, and Audit tables in the database are regularly cleaned of expired and old entries to keep the database size as small as practicable. Entries in the Audit table were cleaned out after 7 days, unless they had been active in a time period equal to twice the average interval; likewise the Ban History table was cleaned when an IP had been lifted for 10 days. Both of these values worked effectively, and in the case of the Audit table reduced its running size by 50%. This prevents any performance gain at the firewall due to the efficiency of the rules from being lost by an overly costly centralised analysis process.

A final addition was made to the system in an effort to ensure that it did not itself become a source of an attack against the systems users in the form of a Denial of Service attack. Through spoofing source IP addresses it would be possible to block someone else's access to a system; whether it is an administrator accessing the network from the outside, or the IP of an external server or application that was required. To prevent a situation like this occurring, a friendly IP table was created in the system. Before any ban is put in place on a gateway, the given IP is checked against the friendly list. If it occurs there then no rule is put in place and the admin is alerted to the situation.

7. Further Work

The implementation of our system has progressed well from being able to initially detect source IP addresses probing multiple gateways to being able to ban them efficiently from the network. As a result of being able to carry out the action needed, the main continuing work on the system will focus on the detection of other modes and types of attack. The greatest challenge in this area in scanner detection is the correlation of scanner activity to draw relationship between different source IP addresses. The goal is to be able to classify users who change their source IP address between scans, enabling them to act with anonymity.

To enable a broader range of detection mechanisms to be developed the authors are building a new system more closely related to the Spice and Spade system. The goal being to initially achieve similar detection rates as both our system and the threshold random walk system, before moving on to develop new methods of detection. The instances of source IPs and their attached events stored within the graph will enlarge the amount of information known about each scanner to facilitate greater detection capability. The continuing work also aims to do this analysis in a distributed fashion, instead of the current centralised approach. The goal being to create a more robust system with built in redundancy, but to also react with greater speed, to possibly facilitate more comprehensive correlation of events.

Amongst the detection goals for the new system are also those of integrating a new method of collecting the data, with the aim of completing this using peer-to-peer technology. Also, we aim to diversify our test bed with data from other sources, from non-continuous IP space, and also from other sensors upon the network, such as Snort.

8. Conclusion

Phase 1 of our system showed that it was possible to detect malicious source IP addresses which were probing multiple gateways connected upon the same network.

Previously such trivial attacks have been overlooked by network security infrastructure, always examining incoming packets in the context of a single gateway, exposing networks to a broader more methodical attack.

Phase 2 allows for such attacks to not only be detected but to also be dealt with through the creation of an Action module which sends out iptables rules to the relevant gateways upon the network, with the aim of providing the needed protection prior to the attacks 'arrival' at the vulnerable gateways.

The Action module not only creates the necessary rules, but also removes the bans once they have expired according to the ban time calculation made by the action module. The result is an efficient attack detection system, aiming to provide protection to ever growing and expanding private networks.

The work is still continuing in scanner detection, but is now moving to a more distributed robust system, with greater correlation aims.

References

[1] Heberlein, L., et al., A Network Security Monitor. Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy, 1990: p. 296-304.

[2] Holden, G., Guide to Network Defence and Countermeasures. Course Technology. 2003: Thomson. 527.

[3] Hofmeyr, S., S. Forrest, and A. Somayaji, Intrusion Detection using Sequences of System Calls. Journal of Computer Security, 1998. 6: p. 151/180.

[4] Kumar, S., Classification and detection of computer intrusions., in Computer Science. 1995, Purdue University. p. 180.

[5] Cheswick, W.R., S.M. Bellovin, and A.D. Rubin, Firewalls and Internet Security Second Edition. Second ed. Professional Computing Series. 2003, Boston: Addison-Wesley. 431.

[6] Porras, P. and P. Neumann. EMERALD: Event Monitoring Enableing Responses to Anomalous Live Disturbances. in National Information Systems Security Conference. 1997. Baltimore, MD.

[7] Sommer, R. and V. Paxson. Enhancing byte-level network intrusion detection signatures with context. in Conference on Computer and Communications Security. 2003. Washington D.C: ACM.

[8] Vigna, G., S.T. Eckmann, and R.A. Kemmerer. The STAT Tool Suite. in DISCEX 2000. 2000. Hilton Head, South Carolina: IEEE Computer Society Press.

[9] Valdes, A. and K. Skinner. Probabilistic Alert Correlation. in Recent Advances in Intrusion Detection : 4th International Symposium. 2001. Davis, CA, USA: Springer Berlin / Heidelberg.

[10] Li, C., Q. Song, and C. Zhang. MA-IDS Architecture for Distributed Intrusion Detection using Mobile Agents. in 2nd International Conference on Information Technology for Application. 2004. Harbin, China.

[11] fyodor, Nmap. 2005.

[12] Zalewski, M., Silence on the Wire: A Field Guide to Passive Reconnaissance and Indirect Attacks. 2005: No Starch Press. 312.

[13] Paxson, V., Bro: a system for detecting network intruders in real-time. Computer Networks, 1999. 31(23--24): p. 2435--2463.

[14] SourceFire, Snort 2.0: Detection Revistited. 2004, SourceFire Inc.: Columbia. p. 7.

[15] Jung, J., et al. Fast Portscan Detection Using Sequential Hypothesis Testing. in IEEE Symposium on Security and Privacy. 2004.

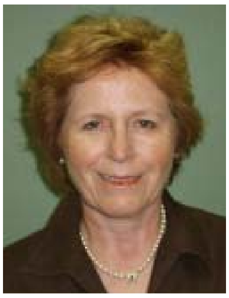
[16] Wald, A., Sequential Analysis. 1947, New York: John Wiley and Sons.

[17] Staniford, S., J.A. Hoagland, and J.M. McAlerney, Practical automated detection of stealthy portscans J. Comput. Secur. , 2002 10 (1-2): p. 105-136

[18] Sipress, A., An Indonesian's Prison Memoir Takes Holy War Into Cyberspace: In Sign of New Threat, Militant Offers Tips on Credit Card Fraud, in Washington Post Foreign Service. 2004: Washington. p. A19.

[19] Hoffman, D., D. Prabhakar, and P. Strooper. Testing iptables. in IBM Centre for Advanced Studies Conference. 2003. Toronto, Ontario, Canada.

Joel Scanlan received his B.Comp. (Hons), from the University of Tasmania, Australia, in 2006. He is 5 currently a PhD candidate. His research interests include Intrusion Detection and Data mining.



Jacqueline Hartnett first degree is a B.A. (Hons) in Geography from Exeter University in the UK and a MComp from the University of Tasmania. After graduating, she worked in the then new computing department of the Royal Dutch Shell group and then for IBM both in Australia and the UK. She has taught computer security courses in the University of Tasmania since 1993. Her current research interests are the use of authentication and access control as a means of maintaining privacy and confidentiality of personally identified data and the development of intrusion detection techniques for groups of collaborating network gateways. She is a member of Australian Standard Review Committees; IT-014 04 System & Data Security, Integrity and Privacy subcommittee, and IT-014-06-08 Electronic Communications in Health working groups. She is also currently chair of the Tasmanian branch of the Australian Computer Society.