InetVis: a Graphical aid for the Detection and Visualisation of Network Scans

Barry V.W. Irwin and Jean-Pierre van Riel

Abstract This paper presents an investigative analysis of network scans and scan detection algorithms. Visualisation is employed to review network telescope traffic and identify incidents of scan activity. Some of the identified phenomena appear to be novel forms of host discovery. The scan detection algorithms of Snort and Bro are critiqued by comparing the visualised scans with alert output. Where human assessment disagrees with the alert output, explanations are sought after by analysing the detection algorithms. The algorithms of the Snort and Bro intrusion detection systems are based on counting unique connection attempts to destination addresses and ports. For Snort, notable false positive and false negative cases result due to a grossly oversimplified method of counting unique destination addresses and ports.

1 Introduction

The Internet is an hostile network environment. Firewalls shelter users from the continual 'storm' of probing activity, commonly referred to as scanning which is, nonetheless, ever present and pervasive throughout the Internet. The greater proportion of this activity is generated by automated malicious software such as worms [1, 2]. While there may be potential value in detecting and tracking scanning activity, it is not a trivial exercise. This paper aims to illustrate that visualisation can better enable the critique of algorithmic scan detection. In particular, two widely deployed open source intrusion detection systems (IDS) are assessed, namely, Snort 2.1.6 [3] and Bro 1.1d [4].

The value of performing scan detection is discussed in the remainder of this introduction. Section 2 highlights the efforts of others with regard to network monitoring, scan detection and network security visualisation. The InetVis visualisation

Barry V.W. Irwin
Rhodes University, Grahamstown, South Africa: b.irwin@ru.ac.za

Jean-Pierre van Riel
Rhodes University, Grahamstown, South Africa: g02v2468@campus.ru.ac.za

tool and key features are described in Section 3. Section 4 discusses the approach used to review network traffic and compare it to IDS scan detection output. It also details the capture of network traffic. The results, which include visual illustrations, are presented in Section 5. The outcomes are concluded in Section 6 with an outline of future work.

1.1 The Merits and Difficulties of Scan Detection

To begin a justification of this research, we need to address the following question, “of what value is scan detection?”

Arguments Against Scan Detection
Firstly, scanning activity is too prevalent and common to warrant concern with each and every individual incident (as detailed further in Section 2.2). Secondly, in production network monitoring scenarios, detecting successful intrusions is of paramount concern whereas scans merely signify vague intent. Thirdly, as stated by several authors [5, 6, 7, 8], current scan detection algorithms have poor accuracy and generate too many false positives. Scan detection is a specialised case of anomaly detection. Many authors argue that anomaly detection methods are less accurate than signature-based methods [9, 10, 11]. Furthermore, algorithms cannot be too complex as they need to be efficient enough for real-time monitoring. For these reasons, scan detection is often left disabled or ignored in production environments.

Arguments For Scan Detection
Having considered reasons advising against scan detection, there at least some motivations for performing scan detection.

Firstly, there is the potential to detect and contain worm activity without relying on signatures. Infected hosts and worm activity can be identified based on the scanning activity they produce, as discussed by Goldi and Hiestand [6]. A general scan detection algorithm could be more scalable and efficient compared to matching traffic against a large signature database.

Scan detection can be employed as an application of 'extrusion' detection – monitoring internal hosts to detect compromised systems that attempt malicious outbound connections [12]. Readily identifying compromised internal hosts can facilitate rapid response and recovery.

There is a similar rationale for performing scan detection on inbound traffic. Both the Snort and Bro IDS have intrusion prevention mechanisms to trigger the injection of new firewall rules as a response to malicious network events. As a pro-active response, once a source is identified as a scanner, subsequent connections can be blocked to prevent future exploit attempts. However, there are at least three caveats to this defence mechanism, namely denial-of-service (DoS), false positives, and distributed attacks. A malicious third party could effect DoS by initiating a scan that spoofs the address of a legitimate host. Similarly, due to the

poor accuracy of scan detection algorithms, benign traffic may be misclassified as scan activity and block legitimate access. The third concern is that distributed attacks from multiple sources will defeat this blocking strategy as the strategy relies upon tracking and blocking individual malicious sources. A more cautious approach would be to maintain a list of 'suspicious' hosts that, on account of their scanning activity, warrant more attention (as alluded to by Verwoerd and Hunt [11]). An IDS can then assign more resources to intensive checks against this reduced set of hosts deemed as suspect.

2 Related Work

The context of this research involves network monitoring methods, intrusion detection theory and information visualisation techniques. This section relates several contributions that the authors believe to be significant.

2.1 Intrusion Detection and the False Positive Problem

One difficulty with intrusion detection is the possibility of falsely identifying legitimate traffic as intrusive. The false positive rate is a major factor that limits the effectiveness of an intrusion detection system. This issue is well addressed by Axelsson *et al* [9]. He takes an established statical argument known as the “base rate fallacy”, and applies it to the problem of intrusion detection. It is presumed that a significant proportion of traffic in a production network is benign, and relative to this, the incidence rate of malicious activity is presumed to be low. Even with high accuracy, a large volume of benign traffic and a low incidence rate of malicious traffic can result in an overwhelming number of false alarms.

2.2 Network Telescopes

Network telescopes can be used to avert the false positive problem as none of the traffic observed by a network telescope is legitimate. Similarly, honeypot networks attempt to capture only malicious activity, but unlike telescopes, actively respond to traffic to solicit more information.

Harder *et al*. provide an example of analysing traffic from a small network telescope [13]. They perform some statical analysis and include some static graphics. A preliminary work to this paper also examines telescope traffic phenomena with emphasis on graphical analysis [14]. However, as stated in a report by Moore *et al*., small network telescopes, such as a class C network, are too small to infer statical generalisations about the Internet [15]. Pang *et al*. perform a large scale study

on a class A and class B networks using both active and passive measurement methods [1]. Lastly, a classical and seminal study by Moore *et al*. approaches the task of inferring denial-of-service (DoS) backscatter [16, 17].

2.3 Classifications of Network Scan Activity

In describing and characterising scan activity, several synonymous terms are used in the literature. This paper adopts the definitions used by Snort documentation [18], as it offers a broad set of categories for classifying scanning activity.

port-scan: a 'one-to-one' scan where a single source host attempts multiple connections to a single target (destination) host on a number of distinct destination ports. This type of scanning is also broadly termed as *service discovery*, or *vertical scanning* [2].

port-sweep: a 'one-to-many' scan on a given destination port where a single host attempts to connect to multiple destination hosts. This can also be referred to as *host discovery*, *address scanning*, *vulnerability scanning* or *horizontal scanning* [2]. Host discovery can also be conducted with the ICMP protocol, as well as merely the IP protocol.

The two above definitions describe probing activity originating from one source alone. To evade detection, both service discovery and host discovery can be coordinated from multiple sources in a distributed manner – referred to as a *distributed scanning*. Snort has the capability to detect distributed port scans (many-to-one). Another deceptive type of scanning is referred to as a *decoy scan*. A single host may spoof multiple source addresses to obscure it's real identity amongst several fake addresses. Lastly, *stealth scans* use a variety of methods to attempt to evade detection. Stealthy techniques include distributed scanning, scanning slowly and using specialised TCP flags (see the nmap reference [19] for more details).

2.4 Algorithmic Approaches to Scan Detection

Scan detection is ultimately a form of anomaly detection. The proficiency of a scan detection algorithm will be determined by how it characterises scan activity and differentiates it from normal traffic. To distinguish between various types of scans requires modelling the distinct characteristics of the traffic patterns produced by each type.

One general assumption is that scan activity will generate a high number of failed connection attempts [20, 18]. Both Snort and Bro apply this as a base assumption in their algorithms [18, 4]. In essence, they simply count failed connection attempts and alert if thresholds are reached. More sophisticated approaches are being developed. For example, advanced statistical approaches can be taken

[5, 7, 21] or data mining classifiers can be employed [8]. For the purposes of this evaluation, the scope is limited to the default algorithms offered by Bro and Snort.

2.5 Network Security Visualisation

Vision is a parallel and pre-attentive cognitive process whereas auditory cognition (used to understand text and speech) is a serial process [22, 23]. Hence, visualisation is a superior medium for correlation and pattern matching tasks. The problem of scan detection can be considered as the task of identifying and characterising anomalous traffic patterns. While visualisation presents these cognitive advantages, scalability is often cited as a limitation [22, 24, 25, 26, 27, 28].

Stephen Lau's 'Spinning Cube of Potential Doom' visualisation is a primary reference for developing this work [20]. The basic 3-D scatter-plot of source IP, destination IP and destination port is well suited to displaying traffic patterns and in particular, scanning activity. One advantage of scatter-plots is that points consume the least amount of display space and hence are the most scalable representation. Lines are a more natural metaphor for connectivity but their use of display space is less efficient.

In building upon Lau's original concept, several other visualisations provided useful ideas. An enumeration of key influences follows. Valdes and Fong discuss a scalable approach with similar plots to the 'Cube of Potential Doom', but in 2-D [24]. The 'space-shield' described by Fisk *et al*. [29] offers features like time-animated replay at variable replay rates and immersive navigation. The parallel axes visualisation by Yin *et al*. [30] discusses focusing on sub-sets of the data, often termed 'drill down'. Their work also includes the concept of a time-window. Ball *et al*. grey-out older events to provide historic context [22]. Etherape highlights the occurrence of new events by momentarily enlarging the thickness of lines [31]. Kuchar *et al*. motivate the importance of time as an attribute for correlating data [32].

3 InetVis Network Traffic Visualisation

InetVis, short for Internet Visualisation, is primarily designed for reviewing network telescope traffic [14]. Fig. 1 illustrates the plotting scheme and basic types of scans. Lau's original visualisation plotted TCP traffic and InetVis extends this by including support for the UDP and ICMP protocols. The input for Lau's visualisation is Bro connection log files. Instead, InetVis uses libpacp to capture live traffic or read packet traces in the libpcap format which is widely used [33]. As described next, InetVis offers a wealth of dynamic and interactive features intended to facilitate exploration of packet capture data.

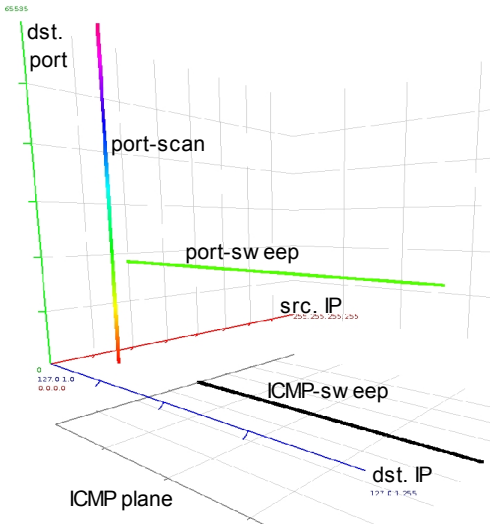

Fig. 1. The InetVis 3-D scatter-plot exhibiting common scan types. Points are plotted according to the source IP (red-axis), destination IP (blue-axis), and destination port (vertical green-axis). TCP and UDP traffic is plotted in the main bounding box and ICMP traffic is plotted to the plane below. For address scanning, a port-sweep appears as a horizontal line, and similarly an ICMP-sweep appears as a line in the ICMP plane. A full port-scan appears as a vertical line. The default is to colour points by destination port with a rainbow colour gradient.

3.1 Key Features and Enhancements

The order and timing of probe packets is of particular interest when characterising network scans. InetVis includes several time-oriented features to enhance the viewers chronological sense of network activity. The replay-position, time-scale and the time-window are 3 features which function closely together. The time-window is relative to the replay position and acts as a filter – older events beyond the time window are excluded. The time-scale adjusts the replay rate to either slow down or speed up playback. In conjunction, these controls manage the time-frame in terms of it's position, size and progression through the steam of packet data. To further enhance chronological salience, transparent ageing fades out older packets. Attention is drawn to the occurrence of a new packet by momentarily enlarging it's point which creates a brief pulse effect. This also helps the viewer notice packets that reoccur.

Other features allow the user to explore, focus and isolate traffic phenomena of interest. Immersive navigation allows the user to move, rotate and zoom the view. A large proportion of traffic targets the lower well-known and registered ports [34]. To spread out the clustering of activity in this region, a logarithmic scale can be applied to the destination port axis (vertical green-axis). The plot can be scaled down into sub-networks or a chosen port range. Packet filtering mechanisms are provided via BPF filter syntax which provides powerful and flexible filtering options. A choice of several colour schemes, such as colouring by source port, source address, packet size and so forth, can aid in viewing attribute relationships in the packet data.

To record visual output InetVis supports capturing snapshots or frame sequences for rendering video clips. InetVis can also output the packets in view to a libpcap format trace file.

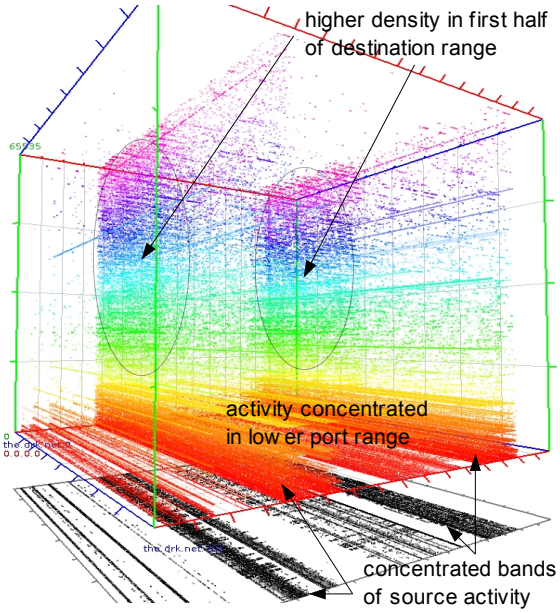

Fig. 2. 6.6 Million packets captured during 2006 from a class C network telescope. Even with the large number of points, InetVis is able to maintain interactive frame rates. Majority of the port-sweeps are concentrated in the lower end of the port range. Note the banding effect by source (red-axis) which shows that a large proportion of activity comes from two sections of the IPv4 address space. Another interesting observation is the higher concentration of scattered activity is the first half of the destination address range (blue-axis).

4 Investigative Methodology

Scanning activity is observed with InetVis, characterised, and compared to the alert output produced by the Snort and Bro. In the first phase of exploration, the network telescope traffic is freely explored with InetVis. In this step, events of interest are discovered, characterised, isolated and recorded. Case-by-case, each event is then processed with Snort and Bro to test the accuracy of the scan detection algorithms. The third investigative phase proceeds in reverse. The full network telescope dataset is processed with Snort and Bro, and a sample of the alerts are reviewed with InetVis. To explain false positive and false negative cases, the source code of the scan detection algorithm is reviewed. Lastly, to verify cases and explanations, specific test traces are created with nmap. Fig. 1 shows an example of three basic scan types generated with nmap. The test cases are used to confirm assessments about how the Snort and Bro scan detection algorithms function.

4.1 Network Telescope Traffic Capture

Traffic captured from network telescope provides a sample of Internet scanning activity. A single monitoring host passively captures traffic destined for the class C network. The dataset consists of monthly packet traces captured during 2006. Due to some downtime (mainly caused by power outages), the data contains a few gaps which amount to 20.2 days (5.5%) for the 2006 year. The accumulated set of

traces for 2006 contains in excess of 6.6 million IP packets. The data composition by protocol and the number of packets is 65% TCP, 20% UDP and 15% ICMP.

The full dataset is visualised in Fig. 2. From this image it is evident that the predominant phenomena are address scans in the form of port-sweeps and ICMP sweeps. Conversely, port-scans are a rarity, as most sources first confirm the presence of a host before expending the time and effort to conduct a port-scan.

4.2 Scan Detection Configuration and Processing

The default Snort and Bro configurations were modified to focus on scan detection features. This streamlined the alert output and avoided unnecessary processing. Several iterations of configuration and testing were performed on the network telescope data as well as generated scan examples.

Snort Configuration
For Snort, only essential preprocessors (flow, frag3, stream4, sfportscan) were loaded to support scan detection. All rule-sets, except 'scan.rules' were disabled. Snort's sfportscan scan detection preprocessor is set to 'low' sensitivity by default. The low setting requires a sub-minimum number of negative responses (termed the 'priority count' threshold [18]). This low setting is inappropriate for analysis with network telescope traffic as the data will not contain any negative responses. Alternatively, the medium and high sensitivity settings do factor in negative responses, but they do not require them. Hence all Snort processing on the data was conducted with either the medium or high sensitivity setting. In the context of a network telescope, disregarding negative responses is acceptable since all of the traffic captured is unsolicited. The sfportscan configuration was also modified to include the detection of ACK scans. The use of ACK scanning will be further discussed in Section 5.2.

Bro Configuration
To focus on scan events, the default Bro 'light' policy was streamlined by removing unnecessary policies intended for application protocol analysis. Bro was used to provide results in two ways. As with Snort, the initial configuration mode was used to test Bro's scan detection with potential false positive and false negative cases. The defaults were tested as well as adaptations to attempt to match the respective threshold options for the medium and high sensitivity levels found in Snort.

The second configuration mode was designed to investigate the distribution of unique addresses targeted by address scans. The bro scan detection policy is highly configurable allowing exact and multiple threshold levels to be specified. This set-up entailed specifying a set of threshold values at regular intervals. Of particular interest is the unique destination address count. As an address scan progresses through the address range, it would trigger an alert at each threshold. A script was written to parse the alert file and count how many scans surpassed each

threshold level. Since a single scan triggers alerts at each threshold it passes, all but the highest alert for that scan should be counted while previous alerts must be discounted. In addition to this, different time thresholds were investigated. Again, unlike Snort's fixed pre-sets, the Bro scan detection time-outs can be redefined to an arbitrary value. Results generated from this are presented in Section 5.1.

4.3 Graphical Exploration and Investigation with InetVis

The network telescope data was explored month by month. To form an overview of all the events in month of traffic, a fast replay rate of 86400x (a day per second) was typically combined with a time window of 7 days. A month's traffic could be skimmed over in roughly 30 seconds. Alternatively, a static view of the full months traffic was used. In this mode, pseudo random patterns formed over long periods could be observed. The overview provided cues on large scale traffic patterns of interest. To reduce the clustering effect in the lower port range, the logarithmic scale was applied to the destination port axis. To look for potential correlations various colour schemes were tested.

Fast replay rates and large time windows were suitable for observing events that progress slowly. The details of faster events to become more evident by gradually reducing the replay rate and time window. Identified events could be focused on by scaling the view and setting the ranges on axes, namely the source sub-network, destination sub-network and destination port range. This 'drills-down' into subsets of the data facilitating a clearer perspective of the event. Further reduction of the time window and replay rate was used to analyse very rapid events. In addition to this, BPF filters [33] can be applied to isolate events.

Once incidents of interest were isolated, they were recorded to capture files for further analysis and testing. The captured files were processed with Snort and Bro. For obvious scans identified with InetVis, the failure to alert indicated a false negative. Furthermore, alert output was inspected to check that detected scans were correctly characterised by the detection algorithm. Alternatively, the Snort and Bro alert logs for each month could be inspected and then investigated with InetVis. Using the alert information to set the appropriate replay position, ranges, and filters eased seeking out the event.

5 Results and Analysis

Network telescope traffic review with InetVis enabled the observation of many anomalous traffic patterns that would not have otherwise be noticed with IDS alert output. The results presented in this section focus on particular findings that illustrate possible flaws in the Snort and Bro scan detection algorithms. Much of the

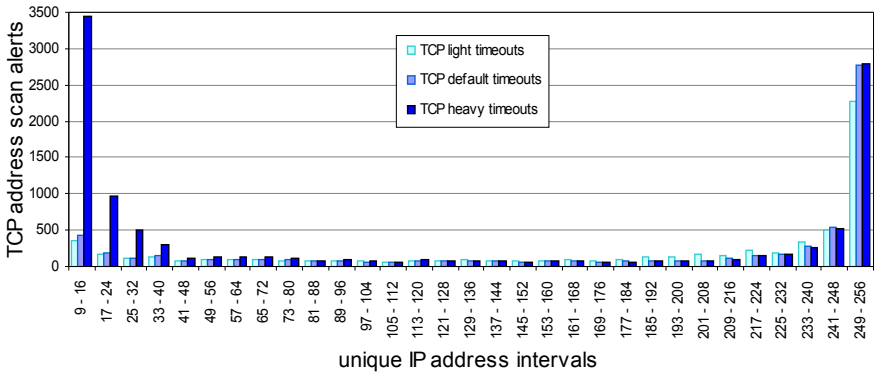

Fig. 3. Number of Bro scan alerts categorised by unique address intervals and time thresholds

discussion entails two attributes used to characterise scans, namely, the unique destination IP address count and the unique destination port count.

5.1 Address Scans and the Distribution of Unique Addresses

The Snort and Bro scan detection algorithms count unique destination ports and addresses. Thresholds for these counts determine scanning activity. Setting appropriate thresholds is an question of parameter optimisation. For Snort, the pre-set thresholds options are conceptually discussed, but how the exact numerical values were arrived at is not specified [18].

With the flexibility afforded by Bro, multiple threshold levels were tested at varied time-out values. The bar chart in Fig. 3 shows the distribution of address scan alerts categorised by the number of unique destination addresses that were targeted. The chart exhibits a full range of thresholds from 9 to 256, grouped by intervals of 8. Essentially, it shows the number of address scans that reached a higher number of unique destination addresses. Furthermore, each address threshold interval is sub-categorised by 'light', 'default' and 'heavy' time-outs. The expirey time-outs are 1 minute, 5 minutes and 10 hours respectively.

Firstly, note the right-hand tail of the distribution in Fig. 3. This shows that the greater proportion of the scanning activity targets almost all the addresses in the class C network telescope. The plot in Fig. 4 expands the density of activity in this upper range for both the TCP and UDP address scans (unfortunately Bro 1.1d does not readily facilitate multiple threshold levels for ICMP). Once again, the greater proportion of scans cover nearly the entire address range. Interestingly, compared to UDP, TCP exhibits this characteristic to a greater extent. Presumably this could be explained in terms of the connection versus connectionless orientations of the protocols.

Returning to Fig. 4, the lower range of the distribution also exhibits a tail, but to a lesser extent. Presumably the tail is caused by miscellaneous non-scan activity. This suggests the obvious. If the IP address threshold is set too low an increas-

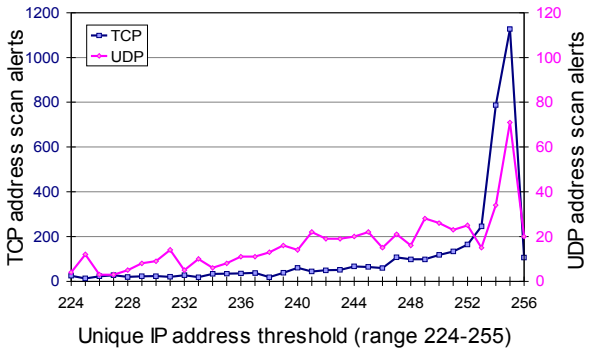

Fig. 4. The number of TCP and UDP address scan alerts plotted for the upper range (224 to 256) of unique destination address thresholds. TCP scale (0-1200) for the count for scan alerts on the left, 0.1x comparative UDP scale (0-120) on the right.

ing number of false-positives will occur. Combined with heavy time-outs, a low UDIP threshold will result in an excessive number of false positives. While lighter time-outs avoid this to some extent, they miss slower stealth scans.

Another difficulty with time-outs is somewhat counter-intuitive at first. One might expect that heavy time-outs would pick-up more scanning activity. However, an astute observation of the upper ranges Fig. 4 shows that for some intervals the 'light' and 'default' values are higher, bar the final interval where the 'default' and 'heavy' values are significantly higher. These offsets can be explained in two ways. Firstly, heavy time-outs might count several individual scan indents as one longer scan. Secondly, if a scan has inconsistent timing between packets, lighter time-outs may miscount the one long scan as two or more smaller scans.

In summary, higher UDIP thresholds will pick-up the majority of scanning activity while avoiding false positives. Time-out thresholds should not be set too long, nor too slow. Clearly, the algorithm needs an improved method of timing activity, so as not to confuse multiple scans as one, or one scan as multiple. A similar issue was discovered with Snort, where it too reports one long scan as multiple shorter scans.

5.2 Scans Discovered and Characterised with InetVis

A multitude of scanning incidents have been identified by human inspection with the InetVis visualisation. From this select examples are presented to illustrate false negative and false positive issues with the Snort and Bro scan detection algorithms.

Pseudo-Random Phenomena

Proportions of the traffic captured from the network telescope exhibit pseudo-random patterns as they are visualised and observed thought InetVis. In general, there are three foreseeable explanations for the pseudo-random phenomena. They are caused by miscellaneous network configuration errors, Denial-of-Service backscatter, or subversive stealthy scanning techniques. Evasive scanning methods employ randomisation, dispersion, patience or a combination thereof. Really slow

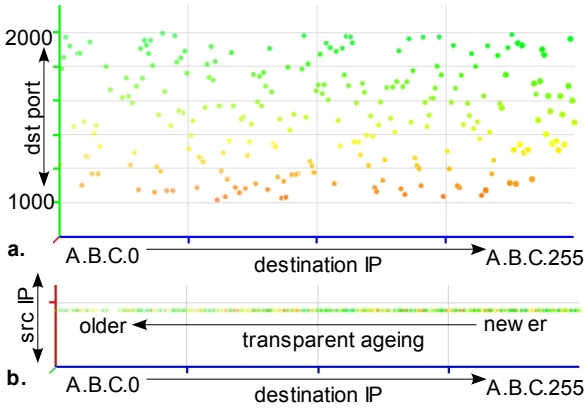

Fig. 5. Rapid 50ms pseudo-random host discovery with probe packets dispersed by destination port. Shown with 75ms time-window, transparent ageing and coloured by destination port. **a.)** Front view showing destination port versus destination IP. **b.)** Top view showing source IP versus destination IP.

scans are likely to fall outside of the bounds of detection time-window thresholds. The authors believe that if a scanning method is sufficiently well dispersed (randomised), it can occur rapidly and still evade detection. An example of this is shown in Fig. 5.

There are two orthographic projections. Fig. 5.a provides a 'frontal' 2-D view, where it exhibits a dramatically fast scatter of packets randomly dispersed about the destination port range from 1000 to 2000. The transparent fading of older packets shows the address range is traversed in a linear progression. Within 50ms, each of the 232 packets targets a unique destination IP. Given the fast transmission rate, there may be a few lost packets. Fig. 5.b shows a top view emphasising how almost the entire address range is covered.

Upon closer inspection, all packets originate from source port 80 and have SYN/ACK TCP flags set. In a normal connection establishment scenario this would be a return packet that indicates that a port is open and accepting a connection. However, the telescope does not initiate any connections. There are two alternative explanations. (1) The observed traffic could be backscatter from a web server undergoing a DoS attack where the telescope's address was spoofed as a source. With DoS attacks, the source addresses are commonly spoofed (faked) to hide the identity of the attacker [16, 17]. (2) As an alternative explanation, it is a form of stealthy host discovery, constituting an address scan.

Supposing the phenomena were backscatter, it would be a curiosity as to why the destination addresses were not randomised and selected from the greater address range of over 4 billion IPv4 addresses. Instead 232 consecutive addresses are probed in a linear fashion – this can be noted from the transparent ageing effect in Fig. 5. The fact that each packet targets a unique address leads the authors toward favouring the address scan explanation.

Although the pseudo-random dispersion in Fig. 5 is not a port-sweep by strict definition, it could be a well adapted alternative to ICMP ping-sweeps. ICMP echo requests and responses are sometimes administratively filtered (by routers and firewalls) to safeguard against attackers who leverage ICMP as a reconnaissance tool. By using TCP source port 80 and setting the TCP flags to SYN/ACK, connection state unaware firewalls will pass this type of traffic. If a destination

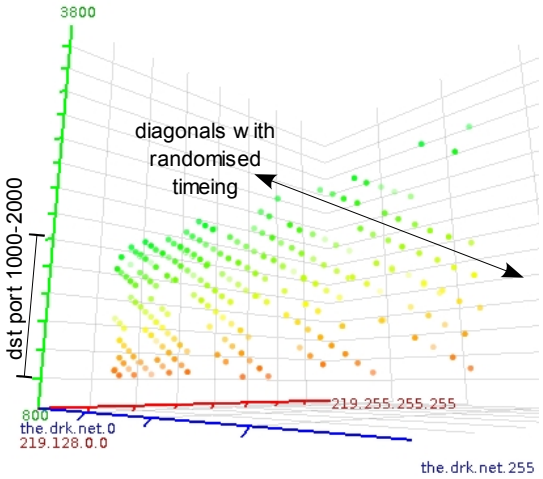

Fig. 6. Pseudo-random diagonal phenomenon dispersed about the destination port range 1000 to 2000. 3-D perspective projection with 36 hour time-window and colour by destination port. Background red axis is source address range scaled to a /9 (half a class A). Foreground blue axis is destination address range – the network telescope's ("the.drk.net"). Vertical green axis is destination port range from 800 to 3800.

host receives an unexpected SYN/ACK packet for a connection it did not initiate, the standard response is to send an RST packet back to the source. Doing so confirms the presence of a host, while no response may indicate that the address is not used (unless the destination network policy does not follow RFC 793 and, similar to the case for ICMP echos, administratively filters out TCP RST packets).

The pseudo-random phenomenon is not an isolated incident. Repeated incidents occur, and several other slower forms have been observed, characteristically bounded in the destination port range of 1000 to 2000. The phenomenon in Fig. 6 is bears some resemblance to that of Fig. 5, but note the more obvious diagonals. The incident occurs in a much longer time frame, 36 hours as opposed to 50ms. Furthermore, the progression across the address range is randomised and repetitive as not every packet targets a unique destination IP address.

The Snort sfportscan algorithm does not alert on the activity in Fig. 5, a possible false-negative. On the contrary, the Bro scan algorithm does alert on this kind of pattern (provided the packets are altered to SYN packets because Bro handles SYN/ACK packets differently). Bro detects this activity because it's algorithm does not consider the destination port when identifying address scans, whereas Snort does. Presumably, this may make it more susceptible to false positives. While Fig. 5 illustrates a somewhat ambiguous case for address scanning, the incident in Fig. 6 is less likely. With extended time-outs, Bro detects Fig. 6 as an address scan whereas snort does not produce any alerts. Whether or not such activity should be classified as address scans or backscatter remains debatable.

Multiple Synchronous Sweep Scans

It is not completely clear whether the examples in Fig. 5 and Fig. 6 aught to be detected as scans. Contrasted to this, a far more obvious case of address scan (port sweep) activity is show in Fig. 7, where 6 simultaneous scans linearly traverse across the address range. As such, this is a multi-vector attack where each port has one or more known vulnerabilities associated with it. Remarkably Snort's sfports-

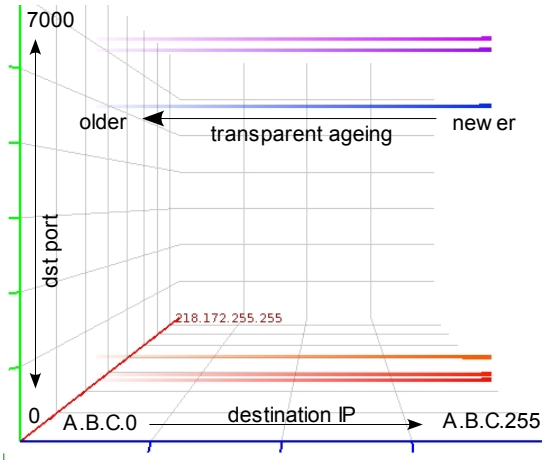

Fig. 7. Multiple simultaneous port sweep. Front-on 3-D perspective projection shown with 150 second time-window and transparent ageing. Colour by destination port. Slightly oversized points at the end of scans highlight the most recent packets. The 6 ports sweeps occur on ports 42 (WINS name service), 139 (SMB/CIFS over NetBios), 445 (SMB/CIFS over TCP), 4899 (radmin windows based remote administration tool), 5900 (VNC remote desktop), 6101 (Verias BackupExec)

can detector produces no alerts for the port sweeps while Bro somewhat misreports the activity.

The Snort False Negative Case

Tested with nmap, Snort is capable of detecting single instances of port-sweeps. For the multi-port-sweep example, it might be assumed that Snort's sfportscan module (scan detection algorithm) would either produce 6 separate port-sweep alerts, or a single alert that encompasses the whole event. Yet no alerts were produced.

Explaining this false negative required review of the Snort source code. The fault is attributed to an over-simplified implementation used to count unique destination addresses and ports. For each source address, instead of maintaining a set of unique destinations, only one previous destination address is kept in memory. The same is true of destination ports. The implementation checks only if the current and previous destinations match, and if not, increments the unique counter. This fails to regard the complete history of destinations within the chosen detection time-window. Effectively, it makes the assumption that an address scan or port-scan will be efficient and not strike the same destination twice.

In the example (Fig. 7), as the multiple port-sweeps progress in simultaneous manner, there is alternation between the 6 ports. This causes repeated hits and the unique port count is continuously incremented instead of remaining at 6. The port count functions as a maximal threshold when detecting address scans. If the unique destination port count is above the threshold, the algorithm rejects the possibility that the traffic pattern constitutes a port-sweep, since traffic allegedly occurs on too many distinct ports – the feature which prevents snort from alerting on the pseudo-random activity in Fig. 5. Thus, a false negative results from over-counting because the algorithm uses both minimal and maximal thresholds to distinguish port-sweeps from port-scans. Similarly this false negative case applies to port-scans too as multiple alternating port-scans will also occur undetected by Snort.

```
Time: 05/30/07-12:45:09.413192
event_id: 30
160.0.0.1 -> 127.0.1.3 (portscan) TCP
Filtered Portsweep
Priority Count: 0
Connection Count: 30
IP Count: 30
Scanned IP Range: 127.0.1.2:127.0.1.3
Port/Proto Count: 1
Port/Proto Range: 32000:32000
```

Fig. 8. Snort sfportscan log output for a false positive case. The test nmap scan only targets 2 addresses, yet the sfportscan unique 'IP count' is in 30. Furthermore, the "IP count' is clearly inconsistent with the 'Scanned IP Range' field which specifies a range of 2 addresses from 127.0.1.2 to 127.0.1.3.

The Bro IDS does produce alert output for the example in Fig. 7 but fails to identify the complete event. In the specific case, it alerted at each set threshold, but only for 1 out of the 6 ports, thereby missing the 5 other scans.

The False Positive Corollary

From the same Snort flaw discussed above, false positives also occur. A false positive can arise if, for a given source, connection attempts repeatedly alternate between two destination addresses, or two destination ports. Instead of recognising the recurring connection attempts to previous destinations, sfportscan continually increments the counters for unique destination addresses and ports. The counter then surpasses a minimal threshold which triggers either a port-sweep or port-scan false alert. As proof of concept for this flaw, a special packet trace was generated with nmap. Multiple alternating TCP SYN connection initiation packets were sent to just two destination addresses on the loop-back interface. The packets were simply alternated 20 times between 127.0.1.2 and 127.0.1.3, totalling 40 packets. This was sufficient to test the pre-sets for the thresholds. Fig. 8 provides and example of sfportscan log output for this false positive. Another observation was that snort logs the event as soon as the threshold is reached thereby failing to report the full extent of a scan.

6 Conclusion

The practical application of visualisation to the problem domain of scan detection has been addressed by this work. InetVis reimplements and extends Stephen Lau's original visualisation concept. It adds several enhancements attuned for visualising network telescope traffic and scanning activity. Special attention is paid chronological salience, allowing the exact order of probing packets to be observed. Several months of network telescope traffic was explored and investigated with InetVis. From this, both false positive and false negative cases were established for the Snort sfportscan module. After further investigation, the inaccuracy was attributed to a unique destination counting flaw. While Bro did not suffer from this flaw, it too failed to accurately report on the full extent of the scan activity, as was the case for the simultaneous port-sweeps in Fig. 7. Pseudo-random phenomena were discussed in Section 5.2 and Fig. 5 could be a stealthy form of host discovery. It

illustrated the difficulty of dealing with ambiguous traffic patterns that could be a form of ACK scanning or backscatter.

InetVis illustrates the advantages of using visualisation and shows how the chosen 3-D scatter-plot is particularly suited to displaying scan activity. Without InetVis, the authors would have a weaker understanding of scan phenomenon and not have discovered these issues with current scan detection algorithms in Snort and Bro.

6.1 Future Work

This work only tests the default scan detection in Snort and Bro. As alluded to in Section 2.4, other scan detection algorithms have been devised. Bro includes and implementation of Jung *et al.*'s algorithm [7] and may offer the flexibility to reimplement and compare other algorithms [5, 8, 21] for future analyses.

To ease the process of conducting visual analysis, the authors are devising semi-transparent visual overlays to represent detected scans. The detection of scans can then be seen against the backdrop of the network traffic. To compliment this, the intention is to include support for reading scan alert output and to automatically forward the replay position to detected scans. In addition, automatic focus of the scan event would be desirable.

One other worthwhile question is to quantitatively assess the performance advantage of Snort's simplified pseudo-unique destination counters. In conjunction, one might look at the accuracy cost of this simplification by judging how many false positives are caused by it. While Bro maintains a set of all previous destinations, this adds complexity and makes the scan detection more resource intensive (in terms of memory consumption and processor time).

References

[1] Pang, Ruoming and Yegneswaran, Vinod and Barford, Paul and Paxson, Vern and Peterson, Larry. Characteristics of internet background radiation. IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement. ACM Press, New York, NY, USA, 2004. p. 27-40.

[2] Yegneswaran, Vinod and Barford, Paul and Ullrich, Johannes. Internet intrusions: global characteristics and prevalence, SIGMETRICS Performance and Evaluation Review, vol. 31, no. 1. ACM Press, New York, NY, USA, 2003. p. 138-147.

[3] Martin Roesch. Snort - Lightweight Intrusion Detection for Networks. LISA '99: Proceedings of the 13th USENIX conference on System administration. USENIX Association, Berkeley, CA, USA,1999. p. 229-238.

[4] Vern Paxson. Bro: a system for detecting network intruders in real-time, Computer Networks, vol. 31, no. 23-24. Elsevier North-Holland, Inc., New York, NY, USA, 1999. p. 2435-2463.

[5] Gates, Carrie and McNutt, Joshua J. and Kadane, Joseph B. and Kellner, Marc I. Scan Detection on Very Large Networks Using Logistic Regression Modeling. Proceedings of the 11th

IEEE Symposium on Computers and Communications (ISCC 2006). IEEE Computer Society, Washington, DC, USA,2006. p. 402-408.

[6] Goldi, Christoph and Hiestand, Roman. Scan Detection Based Identification of Worm-Infected Hosts. Institut fur Technische Informatik und Kommunikationsnetze (Computer Engineering and Networks Laboratory), Eidgenossische Technische Hochschule Zurich (Swiss Federal Institute of Technology Zurich), 2005.

[7] Jung, Jaeyeon and Paxson, Vern and Berger, Arthur W. and Balakrishnan, Hari. Fast Portscan Detection Using Sequential Hypothesis Testing. 2004 IEEE Symposium on Security and Privacy (SP 2006). IEEE Computer Society, Los Alamitos, CA, USA,2004. p. 211-225.

[8] Simon, Gyorgy J. and Xiong, Hui and Eilertson, Eric and Kumar Vipin. Scan Detection: A Data Mining Approach. Sixth SIAM International Conference on Data Mining (SDM06). 2006. p. 118-129.

[9] Axelsson, Stefan. The base-rate fallacy and the difficulty of intrusion detection, ACM Transactions on Information and System Security (TISSEC), vol. 3, no. 3. ACM Press, New York, NY, USA, 2000. p. 186-205.

[10] Kemmerer, Richard A. and Vigna, Giovanni. Intrusion Detection: A Brief History and Overview (Supplement to Computer Magazine), Computer, vol. 35, no. 4. IEEE Computer Society, Los Alamitos, CA, USA, 2002. p. 27-30.

[11] Verwoerd, Theuns and Hunt, Ray. Intrusion detection techniques and approaches, Computer Communications, vol. 25, no. 15. 2002. p. 1356-1365.

[12] Bejtlich, Richard. Extrusion Detection: Security Monitoring for Internal Intrusions. Addison Wesley Professional, 2005.

[13] Harder, Uli and Johnson, Matthew and Bradley, Jeremy T. and Knottenbelt, William J. Observing Internet Worm and Virus Attacks with a Small Network Telescope. Electronic Notes in Theoretical Computer Science, Volume 151, Issue 3, Proceedings of the Second International Workshop on the Practical Application of Stochastic Modeling (PASM 2005). Elsevier, 2005. p. 47-59.

[14] van Riel, Jean-Pierre and Irwin, Barry. InetVis, a visual tool for network telescope traffic analysis. Afrigaph '06: Proceedings of the 4th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa. ACM Press, New York, NY, USA,2006. p. 85-89.

[15] Moore, David and Shannon, Collen and Voelkery, Geoffrey M. and Savagey, Stefan. Network Telescopes: Technical Report. CAIDA, San Diego Supercomputer Center, University of California, San Diego, 2004.

[16] Moore, David and Shannon, Colleen and Brown, Douglas J. and Voelker, Geoffrey M. and Savage, Stefan. Inferring Internet denial-of-service activity, ACM Transactions Computer System (TOCS), vol. 24, no. 2. ACM Press, New York, NY, USA, 2006. p. 115-139.

[17] Moore, David and Voelker, Geoffrey M. and Savage, Stefan. Inferring Internet Denial-of-Service Activity. 10th USENIX Security Symposium. USENIX, ,2001.

[18] Caswell, Brian and Hewlett, Jeremy. Snort Users Manual, version 2.6.1. 2007.

[19] Lyon, Gordon (a.k.a Fyodor). Nmap Reference Guide (Man Page). 2007. http://insecure.org/nmap/man/

[20] Lau, Stephen. The Spinning Cube of Potential Doom, Communications of the ACM, vol. 47, no. 6. ACM Press, New York, NY, USA, 2004. p. 25-26.

[21] Leckie, C. and Kotagiri, R.. A probabilistic approach to detecting network scans. Network Operations and Management Symposium, 2002 (NOMS 2002, IEEE/IFIP). IEEE Computer Society, Washington, DC, USA,2002. p. 359-372.

[22] Ball, Robert and Fink, Glenn A. and North, Chris. Home-centric visualization of network traffic for security administration. VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security. ACM Press, New York, NY, USA,2004. p. 55-64.

[23] Wickens, C. and Sandry, D. and Vidulich, M.. Compatibility and resource competition between modalities of input, central processing, and output., Human Factors, vol. 25, no. 2. 1983. p. 227-248.

[24] Valdes, Alfonso and Fong, Martin. Scalable visualization of propagating internet phenomena. VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security. ACM Press, New York, NY, USA,2004. p. 124-127.
[25] Goodall, John R. and Lutters, Wayne G. and Rheingans, Penny and Komlodi, Anita. Focusing on Context in Network Traffic Analysis, IEEE Computer Graphics and Applications, vol. 26, no. 2. IEEE Computer Society, Los Alamitos, CA, USA, 2006. p. 72-80.
[26] Foresti, Stefano and Agutter, James and Livnat, Yarden and Moon, Shaun and Erbacher, Robert. Visual Correlation of Network Alerts, IEEE Computer Graphics and Applications, vol. 26, no. 2. IEEE Computer Society, Los Alamitos, CA, USA, 2006. p. 48-59.
[27] Erbacher, Robert F. and Christensen, Kim and Sundberg, Amanda. Designing Visualization Capabilities for IDS Challenges. Proceedings of the IEEE Workshops on Visualization for Computer Security (VIZSEC 2005). IEEE Computer Society, Washington, DC, USA,2005. p. 121-127.
[28] Krasser, Sven and Conti, Gregory and Grizzard, Julian and Gribschaw, Jeff and Owen, Henry. Real-time and forensic network data analysis using animated and coordinated visualization. Proceedings from the Sixth Annual IEEE Systems, Man and Cybernetics (SMC) Information Assurance Workshop, 2005.. IEEE Computer Society, Washington, DC, USA,2005. p. 42-49.
[29] Fisk, Mike. and Smith, Steven A. and Weber, Paul M. and Kothapally, Satyarn and Caudell, Thomas P.. Immersive Network Monitoring. Passive and Active Measurement 2003 (PAM2003). 2003.
[30] Yin, Xiaoxin and Yurcik, William and Treaster, Michael and Li, Yifan and Lakkaraju, Kiran. VisFlowConnect: netflow visualizations of link relationships for security situational awareness. VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security. ACM Press, New York, NY, USA,2004. p. 26--34.
[31] Toledo, J. et al. EtherApe: a graphical network monitor. 2006. http://etherape.sourceforge.net/
[32] Kuchar, Olga A. and Hoeft, Thomas J. and Havre, Susan and Perrine, Kenneth A.. Isn't It About Time?, IEEE Computer Graphics and Applications, vol. 26, no. 3. IEEE Computer Society, Los Alamitos, CA, USA, 2006. p. 80-83.
[33] . TCPDUMP Public Repository (tcpdump/libpcap). 2007. http://www.tcpdump.org/
[34] . Port Numbers. 2007. http://www.iana.org/assignments/port-numbers