Using InetVis to Evaluate Snort and Bro Scan Detection on a Network Telescope

B. Irwin and J.-P. 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. Scan detection algorithms used by the Snort and Bro intrusion detection systems are critiqued by comparing the visualised scans with alert output. Where human assessment disagrees with the alert output, explanations are sought by analysing the detection algorithms. The Snort and Bro algorithms 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 a hostile network environment. Firewalls shelter users from the continual "storm" of probing activity (scanning) pervasive throughout the Internet. The greater proportion of this activity is generated by self-propagating malicious software such as worms (Pang et al., 2004; Yegneswaran et al., 2003). While there is value in detecting and tracking scan activity, evaluating the success of automated detection methods can be a complex exercise. This paper aims to illustrate that visualisation can contribute to the critique of algorithmic scan detection. In particular, two widely deployed open source intrusion detection systems (IDS), Snort 2.1.6 (Roesch, 1999) and Bro 1.1d (Paxson, 1999), are assessed.

The remainder of this introduction discusses the value of performing scan detection. Section 2 highlights the efforts of others with regard to network monitoring,

B. Irwin and J.-P. van Riel
Rhodes University, Grahamstown, South Africa, e-mail: b.irwin@ru.ac.za, g02v2468@campus.ru.ac.za

scan detection, and network security visualisation. The InetVis visualisation tool and key features are described in Sect. 3. Section 4 discusses the approach used to review network traffic and compares it to IDS scan detection output. It also describes the samples of network traffic. The results, including visualised examples, are presented in Sect. 5. Future work is outlined in Sect. 6, and the outcomes are concluded in Sect. 7.

1.1 The Merits and Difficulties of Scan Detection

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

1.1.1 Arguments Against Scan Detection

Firstly, scanning activity is too prevalent to warrant concern with every incident (Yegneswaran et al., 2003). Secondly, in production network monitoring scenarios, detecting successful intrusions is of paramount concern, but scans merely signify vague intent. Thirdly, as stated by several authors (Gates et al., 2006; Jung et al., 2004; Simon et al., 2006), 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 (Axelsson, 2000; Kemmerer and Vigna, 2002; Verwoerd and Hunt, 2002). 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.

1.1.2 Arguments for Scan Detection

Having considered arguments against scan detection, there are 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 scan patterns they produce. Compared to matching traffic against a large signature database, a general scan detection algorithm could be more scalable and detect zero-day worm attacks for which no signatures exist. Scan detection can also be employed as an application of "extrusion" detection – monitoring internal hosts to detect compromised systems that attempt malicious outbound connections (Bejtlich, 2005). Readily identifying compromised internal hosts can facilitate rapid response and recovery.

There is a 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 affect 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, blocking legitimate access. The third concern is that distributed attacks from multiple sources will defeat this blocking strategy, as it relies upon tracking and blocking individual malicious sources. A more cautious approach would be to maintain a list of "suspicious" hosts that, due to their scanning activity, warrant more attention, as alluded to by Verwoerd and Hunt (2002). An IDS can then assign more resources to intensive checks against this reduced set of hosts deemed suspect by virtue of their previous scanning activity.

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 IDSs, an issue is well addressed by Axelsson (2000). He takes an established statistical 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 low. Even with high accuracy, a large volume of benign traffic and a low incidence of malicious traffic can result in an overwhelming number of false alarms.

2.2 Network Telescopes

Network telescopes are constituted by segments of unassigned IP address space where unsolicited traffic is passively captured. This averts the false positive problem because legitimate traffic will never be observed. The observations tend to be limited to scans and backscatter, because network telescope IP addresses do not initiate or respond to traffic. Similarly, honeypot networks attempt to capture only malicious

activity but, unlike telescopes, actively respond to traffic to solicit more information and improve the scope of the observations.

Harder et al. (2005) provide an example of analysing traffic from a small network telescope. They perform some statistical analysis and include some static graphics. In preliminary work, the authors (van Riel and Irwin, 2006), examine telescope traffic phenomena with an emphasis on interactive graphical analysis. However, Moore et al. (2004) argue that small network telescopes, such as a class C network, are too small to infer statistical generalisations about the Internet. Pang et al. (2004) perform a large-scale study on class A and B networks using both active and passive measurement methods. Lastly, using large telescopes, a seminal study by Moore et al. (2006) discusses the task of inferring DoS backscatter.

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 (Caswell and Hewlett, 2007), as it offers a broad set of categories for classifying scanning activity:

port-scan: a "one-to-one" scan where a 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 *service discovery*, or *vertical scanning* (Yegneswaran et al., 2003).

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* (Yegneswaran et al., 2003). Host discovery can also be conducted with ICMP.

These 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 *distributed scanning*. Snort has the capability to detect distributed and decoy port scans (many-to-one). A single host may spoof multiple source addresses as decoys to obscure its real identity. 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 (Lyon, 2007) for more details.

2.4 Algorithmic Approaches to Scan Detection

Scan detection is 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. Distinguishing between various scan types requires modelling the distinctive characteristics of traffic patterns produced by each type. One general assumption is that scan activity will generate a high number of failed connection

attempts (Lau, 2004; Caswell and Hewlett, 2007). Both Snort and Bro apply this as a base assumption in their algorithms. In essence, they simply count failed connection attempts and alert if thresholds are reached, though more sophisticated approaches are being developed. For example, advanced statistical approaches can be taken (Gates et al., 2006; Jung et al., 2004; Leckie and Kotagiri, 2002), or data mining classifiers employed (Simon et al., 2006). 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 (Ball et al., 2004; Wickens et al., 1983). Hence, visualisation is a superior medium for correlation and pattern matching tasks such as observing anomalous traffic patterns caused by scans. However, despite these cognitive advantages, scalability is often cited as a limitation (Ball et al., 2004; Valdes and Fong, 2004; Goodall et al., 2006; Foresti et al., 2006).

Stephen Lau's "Spinning Cube of Potential Doom" visualisation is a primary reference for developing this work (Lau, 2004). 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. Scatter-plots have a scalability advantage because points consume a minimal amount of display space. Lines are commonly used in network visualisation (Ball et al., 2004; Yin et al., 2004; Toledo et al., 2006) to provide a natural metaphor for connectivity, but use display space less efficiently.

In building on 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 (Valdes and Fong, 2004). The "space-shield" described by Fisk et al. (2003) offers features like time-animated replay at variable replay rates, and immersive navigation. The parallel axes visualisation by Yin et al. (2004) discusses focusing on subsets of the data, often termed "drilling down". Their work also includes the concept of a time-window (Ball et al., 2004) grey-out older events to provide historic context. Etherape highlights the occurrence of new events by momentarily enlarging the thickness of lines (Toledo et al., 2006). Lastly, Kuchar et al. (2006) motivate the importance of time as an attribute for correlating data.

3 InetVis Network Traffic Visualisation

InetVis, short for Internet Visualisation, is primarily designed for reviewing network telescope traffic (van Riel and Irwin, 2006). Figure 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, whereas InetVis captures live traffic

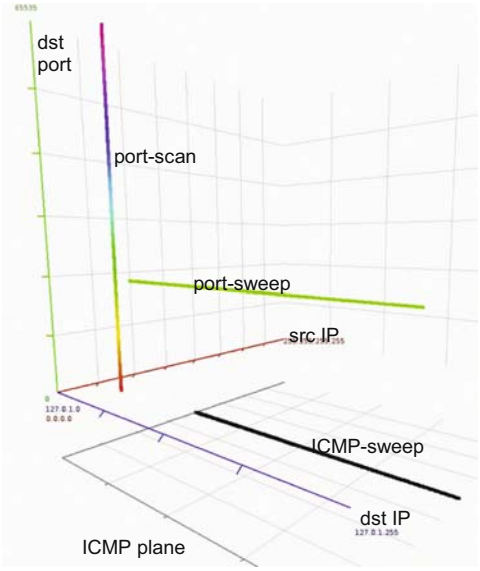

Fig. 1 The InetVis 3-D scatter-plot, exhibiting common scan types. Points are plotted according to the source IP (*red-axis*), destination IP (*blue-axis*), and destination port (*vertical green-axis*). TCP and UDP traffic is plotted together in the main bounding box and ICMP traffic is plotted to the 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. These example scans were generated with Nmap – for more Nmap scan examples, refer to previous work in (van Riel and Irwin, 2006). The default is to colour points by destination port with a rainbow colour gradient

or reads packet traces in the common Libpcap format (supported in Snort, Bro, Tcpdump, and Wireshark). InetVis offers a wealth of dynamic and interactive features intended to facilitate exploration of packet capture data.

3.1 Key Features and Enhancements

When characterising network scans, the order, and timing of probe packets is significant. InetVis includes several features to enhance the viewer's chronological sense of network activity – the time-window, replay-position, time-scale, transparent ageing, and pulse features. The time-window is relative to the replay position and acts as a filter by excluding events that are before or beyond the bounds of the window. The time-scale adjusts the replay rate to either slow or speed up playback. In conjunction, these controls manage the time frame in terms of position, size, and progression through the steam of packets.

Each feature can be independently controlled to dynamically adjust the time frame. For each control, quick adjustments can be made by slider-bars that scale the adjustment effect to the appropriate range of time. For example, in lower ranges, adjustments of the time-scale and time-window are in the order of milliseconds and seconds for fine control. Conversely, the upper ranges are adjusted in the order of hours (for the time-scale) or days, weeks, and months (for the time-window).

Transparent ageing and pulse effects can be enabled to enhance chronological salience. With transparent ageing, older events are faded out and appear diminished in contrast to newer events that appear solid (fully opaque). This creates an emphasis on newer events while maintaining a lingering sense of historic context. Enabling the brief pulse effect draws attention to newly introduced events by enlarging their points momentarily. This also helps the viewer to notice reoccurring events.

Other features allow the user to explore, isolate, and focus on interesting traffic phenomena. These features entail navigation, scaling the plot, a logarithmic plot, filtering, colour schemes, and recording output. Immersive navigation allows the user to explore within the data by moving, rotating, and zooming the view in 3-D space. The user can form additional insight into subtle patterns by viewing the data from different perspectives. To avoid disorientation, the navigation is bounded and ensures that the user cannot lose sight of the visualisation.

To explore the data in more detail, and deal with the effects of over-plotting (where points overlap each other), the plot can be scaled down into sub-networks or a smaller port range. Source and destination networks are specified in CIDR notation and the destination port range is specified by an upper and lower bound. By default, the plots linearly map the chosen network and port ranges onto their respective axes. For the destination port axis (vertical green-axis), a logarithmic scale can be applied instead of linear mapping. This resolves over-plotting that occurs at lower range due to the high proportion of traffic that targets the well-known and registered ports. For fine control, the user can adjust the base of the logarithm to control the amount of expansion at the lower port region – the greater the base, the greater the expansion at lower ports, with the effect of more contraction at higher ports.

Uninteresting traffic can be filtered with BPF syntax (as used in Tcpdump). The syntax is complicated, but facilitates powerful and flexible filtering options that can operate on any field in the packet data. The filters are applied dynamically by rereading the packet trace data (with the caveat that large capture files incur some delay in applying the change). Colour can convey additional information and the user has a choice of several colour schemes, such as colouring by source port, source address, packet size, protocol, and so forth. Colour change interactions are dynamic, and can aid in viewing attribute relationships in the packet data.

To record output, InetVis supports capturing image snapshots or frame sequences for rendering video clips. The packets in view can also be dumped to a trace file in Libpcap format, allowing the user to review traffic with other utilities, such as Tcpdump and Wireshark.

4 Investigative Methodology

Scanning activity is observed with InetVis, characterised, and compared to 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 samples 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 a 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 establish the presence of a host before expending the time 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.

4.2.1 Snort Configuration

For Snort, only essential pre-processors (`flow, frag3, stream4, sfportscan`) were loaded to support scan detection. All rule-sets, except `scan.rules` were disabled. Snort's `sfportscan` scan detection pre-processor is set to "low" sensitivity by default. The low setting requires a sub-minimum number of negative responses, and is named the "priority count" threshold (Caswell and Hewlett, 2007). 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.

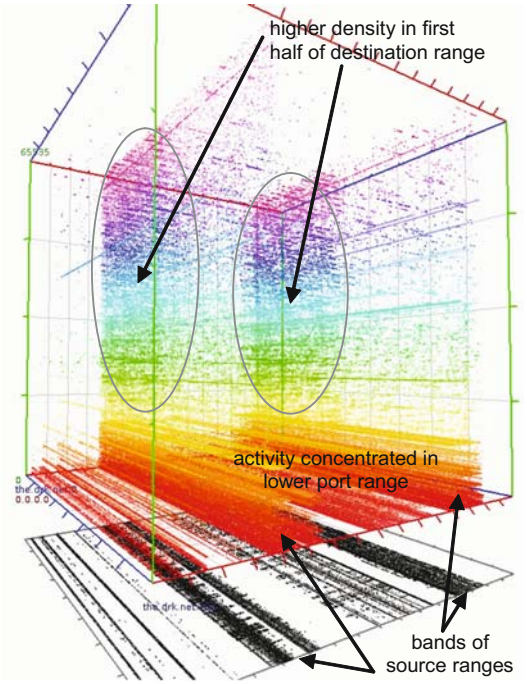

Fig. 2 6.6 Million packets captured during 2006 from a class C network telescope. Even with the large number of points, InetVis is able to maintain moderately interactive frame rates using mid-range graphics hardware (e.g. 8 FPS with a Nvidia GeForce 7600GS). The majority of port-sweeps are concentrated 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 in the first half of the destination address range (*blue-axis*)

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 (discussed further in Sect. 5.2.)

4.2.2 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 setup entailed specifying a set of threshold values at regular intervals. Of particular interest is the unique destination address count. As an address scan progressed through the address range, it triggered an alert at each threshold. A script was written to parse the alert file, counting 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, different time thresholds were investigated. Unlike Snort's fixed pre-sets, Bro scan detection time-outs can be redefined to an arbitrary value. Results generated from this are presented in Sect. 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, a fast replay rate of $86,400\times$ (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 s. Alternatively, a static view of all the traffic was viewed with a 30-day time window. This mode suited the observation of slow scans and pseudo-random patterns formed over long periods. Patterns were identified as pseudo-random when some facet of non-randomness could be noticed – for example, scattered packets that ended up forming diagonal lines (as discussed later in Sect. 5.2). To reduce the clustering effect in the lower port range, the logarithmic scale was applied to the destination port axis. Various colour schemes were tested when searching for possible correlations.

Rapid replay rates and large time windows were suitable for observing events that progressed slowly. The details of fast events became more evident by 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 "drilled down" into subsets of the data, facilitating a clearer perspective of the event. By reducing the range of data viewed, smaller scale events became more evident. Further reduction of the time window and replay rate was used to analyse very rapid events. In addition, BPF filters were applied to isolate events of interest.

Once incidents 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 could not 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 discussion entails two attributes used to characterise scans; the unique destination IP address count, and the unique destination port count.

5.1 Address Scans and the Distribution of Unique Addresses

Snort and Bro scan detection algorithms count unique destination ports and addresses. A combination of thresholds determine if scanning activity has occurred and setting appropriate threshold levels is a question of parameter optimisation. Snort's "high", "medium", and "low" sensitivity pre-sets are hard-coded threshold combinations with limited scope for optimisation.

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 expiry time-outs are 1 min, 5 min, and 10 h, 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.

Returning to Fig. 3, 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 – setting the IP address threshold too low increases the number of false-positives. Combined with heavy time-outs, a low unique destination

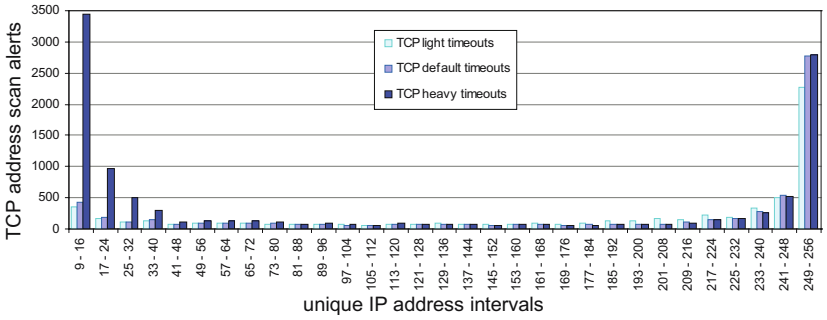

Fig. 3 Number of Bro scan alerts categorised by unique address intervals and time thresholds

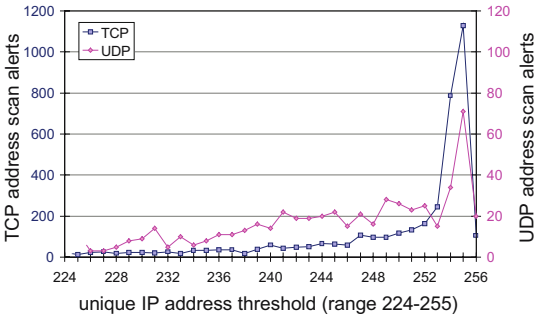

Fig. 4 The number of TCP and UDP address scan alerts are plotted (*y*-axis) for the upper range 224–256 of unique destination address thresholds (*x*-axis). The count for TCP scan alerts count range is on the left, and ranges between 0 and 1,200. For UDP on the right, the range is 0–120 (10× smaller than TCP)

IP threshold will result in excessive false positives. While lighter time-outs avoid this to some extent, they miss slower stealth scans.

Another difficulty with time-outs seems somewhat counter-intuitive initially. One might expect that heavy time-outs would pick up more scanning activity. However, careful observation of the upper ranges in Fig. 3 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: either, heavy time-outs count several individual scan incidents as one longer scan, or, if a scan's timing between packets is inconsistent, lighter time-outs miscount one long scan as two or more smaller scans.

In summary, higher unique destination IP 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, which also reports one long scan as multiple shorter scans.

5.2 Scans Discovered and Characterised with InetVis

Multitudes of scanning incidents have been identified by human inspection with InetVis. Selected examples are presented to illustrate false negative and false positive issues with the Snort and Bro scan detection algorithms.

5.2.1 Pseudo-Random Phenomena

Some traffic captured from the network telescope exhibited pseudo-random patterns when visualised with InetVis. In general, there are three foreseeable explanations for the pseudo-random phenomena. They are caused by miscellaneous network configuration errors, DoS backscatter, or subversive stealthy scanning techniques. Evasive scanning methods employ randomisation, dispersion, patience or a combination

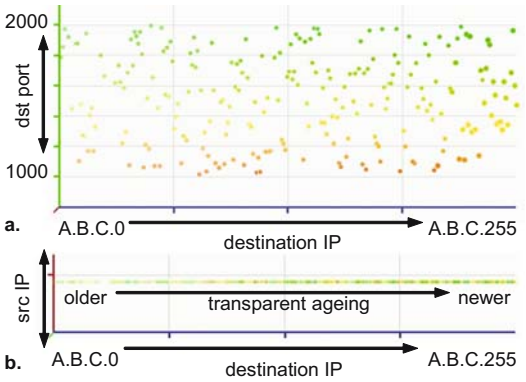

Fig. 5 Rapid 50 ms pseudo-random host discovery with probe packets dispersed by destination port. The image is shown with a 75 ms time-window, transparent ageing, point-pulse for new events, and coloured by destination port. (**a**) Front-view showing destination port vs. destination IP. (**b**) Top-view showing source IP vs. destination IP

thereof. Very slow 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 while evading detection, as shown in Fig. 5.

There are two orthographic projections. Figure 5a provides a frontal 2-D view exhibiting a dramatically fast scatter of packets randomly dispersed about the destination port range 1,000–2,000. The transparent fading of older packets shows the address range is traversed in a linear progression. Within 50 ms, each of the 232 packets targets a unique destination IP. Given the fast transmission rate, a few packets may have been lost. Figure 5b 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. Assuming the normal connection establishment procedure, a SYN/ACK response indicates that the port is accepting connections. However, the telescope does not initiate any connections. Two explanations are raised: (1) The observed traffic could be backscatter from a web server undergoing a DoS attack where the telescope's source address was forged (spoofed) – DoS attacks commonly spoof source addresses to hide the identity of the attacker (Moore et al., 2006). (2) Alternatively, this is a form of stealthy host discovery, constituting an address scan.

Supposing the phenomena were backscatter, it is curious that the spoofed 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. Added to this, each packet targets a unique address, and this leads the authors to favour 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 host receives

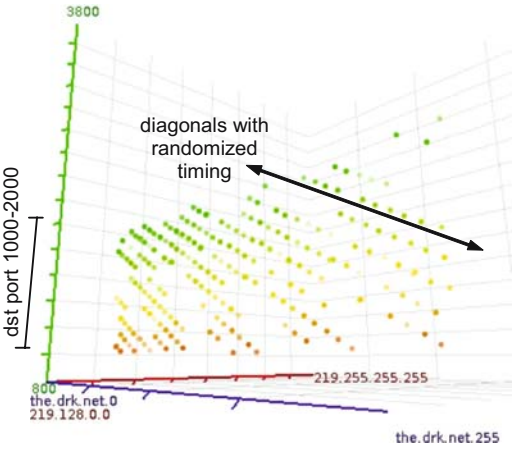

Fig. 6 Pseudo-random diagonal phenomenon dispersed about the destination port range 1,000–2,000. The view is in 3-D perspective projection with a 36 h time-window and coloured by destination port. The *red axis* in the background represents the source address range, which is scaled to a/9 network block (half a class A). In the foreground, the *blue axis* represents the destination address range – the network telescope's range ("the.drk.net"). The *vertical green axis* represents the destination port range from port 800 to 3,800

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 echoes, 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 1,000–2,000. The phenomenon in Fig. 6 bears some resemblance to that in Fig. 5, but note the obvious diagonals. The incident occurs in a much longer time frame, 36 h rather than 50 ms. 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. By contrast, the Bro scan algorithm does alert on this kind of pattern (provided the packets are altered to SYN packets, as Bro handles SYN/ACK packets differently). Bro detects this activity because, unlike Snort, its algorithm does not consider the destination ports when identifying address scans. Arguably, this leaves 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.

5.2.2 Multiple Synchronous Sweep Scans

While it is not completely clear if the examples in Figs. 5 and 6 should be detected as scans, a far more obvious case of address scan (port-sweep) activity is shown in

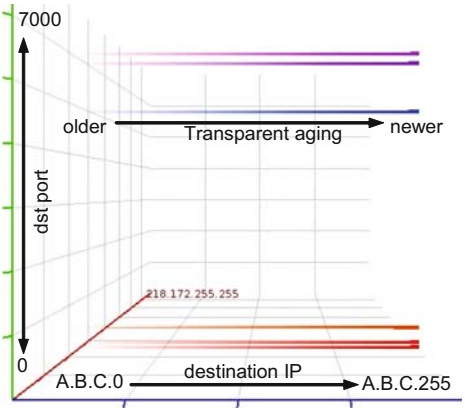

Fig. 7 Simultaneous port-sweeps. Front-on 3-D perspective projection shown with 150 s time-window and transparent ageing. Colour by destination port. Slightly oversized points at the end of scans highlight the most recent packets. The six port-sweeps occur on ports 42 (WINS name service), 139 (SMB/CIFS over NetBios), 445 (SMB/CIFS over TCP), 4,899 (radmin windows based remote administration tool), 5,900 (VNC remote desktop), 6,101 (Verias BackupExec)

Fig. 7, where six simultaneous scans linearly traverse the address range. This is a multi-vector attack, where each port is associated with one or more vulnerabilities. Remarkably, Snort's `sfportscan` detector produces no alerts for the port-sweeps, while Bro somewhat misreports the activity.

5.2.3 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 would produce either six separate port-sweep alerts, or a single alert for the whole event. Yet no alerts were produced.

Explaining this false negative required review of Snort's source code. The fault is attributed to an over-simplified implementation for counting unique destination addresses and ports. For each source address, instead of maintaining a set of unique destinations, only the previous destination address is kept in memory, and a similar case applies to tracking destination ports. The current destination is compared with just the previous destination, and increments the unique counter if they do not match. This fails to consider the complete history of destinations within the chosen detection time-window. Effectively, it makes the poor assumption that a port-sweep or port-scan will be efficient and not strike the same destination twice.

In Fig. 7, as multiple port-sweeps progress simultaneously, there is alternation between six ports. This causes repeated hits, and the unique port count is continuously incremented instead of remaining at six. 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 of a port-sweep, since traffic apparently occurs on too many distinct ports – the feature that 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

```
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 targeted two addresses, yet the unique "IP count" is 30. Furthermore, the "IP count" is clearly inconsistent with the "Scanned IP Range" field which specifies a range of two addresses from 127.0.1.2 to 127.0.1.3

thresholds to distinguish port-sweeps from port-scans. This false negative case also applies to port-scans, as multiple alternating port-scans will go undetected by Snort.

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 one out of the six ports, thereby missing the five other scans.

5.2.4 The False Positive Corollary

The Snort flaw discussed above also generates false positives. 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. Figure 8 provides an 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 Future Work

This work tested the default scan detection in Snort and Bro. As alluded to in Sect. 2.4, other scan detection algorithms have been devised. Bro includes an implementation of the algorithm by Jung et al. (2004) and may offer the flexibility to re-implement and compare other algorithms (Gates et al., 2006; Simon et al., 2006; Leckie and Kotagiri, 2002) 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 complement this, the intention is to include support for reading scan alert output and automatically forward the replay position to detected scans. Automatic focus of the scan event would also be desirable.

Another worthwhile investigation would quantitatively assess the performance advantage of Snort's simplified pseudo-unique destination counter. In conjunction, one might look at the accuracy cost of this simplification by judging how many false positives it generates. 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). Of course, this kind of analysis should make use of production traffic to test real-world performance.

7 Conclusion

This work exhibits the practical application of visualisation to the problem domain of scan detection. InetVis re-implements and extends Stephen Lau's original visualisation concept, adding several enhancements for visualising network telescope traffic and scanning activity. Special attention is paid to chronological salience, allowing the exact order of probing packets to be observed. Several months of network telescope traffic were explored and investigated with InetVis. From select scan incidents, false positive and false negative cases were established for the Snort `sfportscan` module, and this inaccuracy was attributed to a unique destination-counting flaw. While Bro did not suffer from this flaw, it too failed to report the full extent of scan activity, as shown with simultaneous port-sweeps in Fig. 7. Pseudo-random phenomena were discussed in Sect. 5.2, and Fig. 5 could be a stealthy form of host discovery. It illustrates the difficulty of dealing with ambiguous traffic patterns that could be a form of ACK scanning or backscatter.

InetVis showcases the advantages of using visualisation, and the 3-D scatter-plot proves to be a suitable choice for displaying scan activity. Without InetVis, the authors would have a weaker understanding of scan phenomena, and would not have discovered these issues in the Snort and Bro scan detection algorithms.

References

Axelsson S (2000) The base-rate fallacy and the difficulty of intrusion detection. ACM Transactions on Information and System Security 3(3):186–205

Ball R, Fink GA, North C (2004) 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, 55–64. ACM Press, New York, USA

Bejtlich R (2005) Extrusion Detection: Security Monitoring for Internal Intrusions. Addison-Wesley, Reading, MA

Caswell B, Hewlett J (2007) Snort Users Manual, Version 2.6.1. http://www.snort.org/docs/snort_manual.pdf. Accessed 3 April 2007

Fisk M, Smith SA, Weber PM et al (2003) Immersive network monitoring. PAM '03: 2003 Passive and Active Measurement Conference. http://woozle.org/~mfisk/papers/pam03.pdf. Accessed 14 November 2007

Foresti S, Agutter J, Livnat Y et al (2006) Visual correlation of network alerts. IEEE Computer Graphics and Applications 26(2):48–59

Gates C, McNutt, JJ, Kadane JB et al (2006) Scan detection on very large networks using logistic regression modeling. ISCC '06: Proceedings of the 11th IEEE Symposium on Computers and Communications, 402–408. IEEE Computer Society, Washington, DC, USA

Goodall JR, Lutters WG, Rheingans P et al (2006) Focusing on context in network traffic analysis. IEEE Computer Graphics and Applications 26(2):72–80

Harder U, Johnson M, Bradley JT et al (2005) Observing internet worm and virus attacks with a small network telescope. PASM '05: Proceedings of the Second International Workshop on the Practical Application of Stochastic Modelling. ENTCS, 151(3):47–59. doi:10.1016/j.entcs.2006.03.011

Jung J, Paxson V, Berger AW et al (2004) Fast portscan detection using sequential hypothesis testing. SP '04: Proceedings of the 2004 IEEE Symposium on Security and Privacy, 211–225. IEEE Computer Society, Los Alamitos, CA, USA

Kemmerer RA, Vigna G (2002) Intrusion detection: a brief history and overview (supplement to Computer Magazine). Computer 35(4):27–30

Kuchar OA, Hoeft TJ, Havre Susan et al (2006) Isn't it about time? IEEE Computer Graphics and Applications 26(3):80–83

Lau S (2004) The spinning cube of potential doom. Communications of the ACM 47(6):25–26

Leckie C, Kotagiri R (2002) A probabilistic approach to detecting network scans. NOMS '02: Network Operations and Management Symposium, 359–372. IEEE Computer Society, Washington, DC, USA

Lyon G (2007) Nmap reference guide (man page). http://insecure.org/nmap/man/. Accessed 11 June 2007

Moore D, Shannon Collen, Voelker GM et al (2004) Network telescopes: technical report. CAIDA, San Diego. http://www.caida.org/publications/papers/2004/tr-2004-04/tr-2004-04.pdf. Accessed 10 April 2007

Moore D, Shannon C, Brown DJ et al (2006) Inferring internet denial-of-service activity. ACM Transactions Computer System 24(2):115–139

Pang R, Yegneswaran V, Barford P et al (2004) Characteristics of internet background radiation. IMC '04: Proceedings of the Fourth ACM SIGCOMM Conference on Internet Measurement, 27–40. ACM Press, New York, USA

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

Roesch M (1999) Snort – lightweight intrusion detection for networks. LISA '99: Proceedings of the 13th USENIX Conference on System Administration, 229–238. USENIX Association, Berkeley, CA, USA

Simon GJ, Xiong H, Eilertson E, et al (2006) Scan detection: a data mining approach. SDM '06: Proceedings of the Sixth SIAM International Conference on Data Mining, 118–129. SIAM, Philadelphia, USA

Toledo J et al (2006) EtherApe: A Graphical Network Monitor. http://etherape.sourceforge.net/. Accessed 23 April 2006

Valdes A, Fong M (2004) Scalable visualization of propagating internet phenomena. VizSEC/DMSEC '04: Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security, 124–127. ACM Press, New York, USA

van Riel J-P, Irwin BV (2006) InetVis, a visual tool for network telescope traffic analysis. Afrigaph '06: Proceedings of the Fourth International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa, 85–89. ACM Press, New York, USA

Verwoerd T, Hunt R (2002) Intrusion detection techniques and approaches. Computer Communications 25(15):1356–1365

Wickens C, Sandry D, Vidulich M (1983) Compatibility and resource competition between modalities of input, central processing, and output. Human Factors 25(2):227–248

Yegneswaran V, Barford P, Ullrich J (2003) Internet intrusions: global characteristics and prevalence. SIGMETRICS Performance and Evaluation Review 31(1):138–147

Yin X, Yurcik W, Treaster M, et al (2004) 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, 26–34. ACM Press, New York, USA