Detection of network scan attacks using flow data

Joris Kinable
j.kinable@student.utwente.nl

ABSTRACT

Current Network Scan Detection Systems (NSDS), usually implement detection schemes which depend on the ability to analyze every single network packet in detail. In order to scale NSDS to high speed networks, processing gigabits every second, a different approach is required since packet level inspection is no longer feasible.
In this paper we will investigate the possibilities of using netflow data, comprising an aggregation of the information contained in multiple packets, as a means to detect network scanners. The usage of netflow data imposes restrictions on the detection approaches since detailed packet information is lost. The main contribution of this paper is the identification of detection approaches applicable in high speed networks. The approaches elaborated generalize the ideas behind conventional detection approaches. In addition, a new detection approach is added, based on observed connection patterns. To analyze the results achieved while putting our detection approaches into practice, a set of real-life netflow records is used. Final validation of the results is performed by comparing the results of distinctive detection approaches mutually. It turns out that, although in many cases the information in the netflow records is not sufficient to identify scan attempts with absolute certainty, the approaches are quite capable of filtering out a set of suspicious hosts.

Keywords

High-speed network, Netflow data, Network security, Scan detection

1. INTRODUCTION

With the ongoing expansion of the internet and its services, network security aspects become increasingly essential. In this perspective, the detection of network attacks plays an important role. One of the major challenges for security management consists of developing systems that detect attempts to scan networks, as attackers generally start with a reconnaissance phase, trying to characterize the hosts or networks they are considering hostile activity against. However, further research is needed to improve the automatic detection of these abnormal network access patterns,.

Although previous research has already greatly improved our insight into network scan detection systems (NSDS) and their application, their detection scheme usually analyzes individual network packets. In a relatively low speed network of approximately 100 Mbit up to 1 Gbit [1], [2], processing and validating every distinct network packet is still possible indeed. For high speed networks, however, this is hardly feasible and different standards and approaches are required. Detection of network scan attacks in high speed networks is the focus of this paper. Due to the large amounts of data crossing these networks, analyzing all network packets in detail is no longer an option. Therefore our analysis will use flow data in order to detect network anomalies caused by scan attacks.

The main question we examine in this paper is whether it is possible to use network scan detection approaches to operate on network flow data of high speed networks. This problem statement will be supported by three sub questions:

1. Which scan attack techniques and detection approaches are identified by literature?
2. Which detection approaches can be used for high speed networks?
3. Which results are achieved when applying the proposed detection approaches in practice?

The first question will be dealt with in chapter 2, providing a literature based overview of scan attack and detection techniques. The second question will be covered in chapter 3. Developing detection approaches for anomalous traffic patterns will be pursued in the following two directions. First, the development will mainly result from the generalization and aggregation of features of conventional detection approaches, gathered from the consulted works in chapter two. Second, an additional approach will be presented, using a type of detection algorithm that drew little attention in the conventional detection approaches so far.
Chapter 4 corresponds to the third and last of the subquestions enumerated before. Here we validate the proposed algorithms in practice, and investigate whether they do actually contribute to the detection of scanners. The behaviour of the proposed detection algorithms is evaluated by using a set of real-time netflow data existing of 3 hours of university network traffic (class B network), taking into consideration approximately 23 million TCP network connections. Discoveries of scan attacks will be verified by comparing the results of distinctive detection approaches mutually. Chapter 5 finally offers the conclusion and suggestions for further research.

2. CONVENTIONAL ATTACK TYPES AND DETECTION APPROACHES

This chapter provides a concise overview of scan types, evasive techniques and detection approaches identified in literature. The information incorporated has been gathered through extensive search in conference proceedings, Google Scholar, library search, Wikipedia and references in state-of-the-art literature.

9th Twente Student Conference on IT, Enschede 23th June, 2008

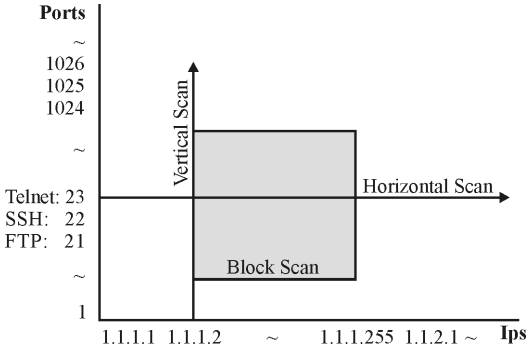

Figure 1: Scan types

2.1 Scan attacks and evasion techniques

Traditionally, network scanners are classified by literature according to their attack pattern (Figure 1) in horizontal scanners, targeting a single port on several systems, vertical port scanners probing multiple ports on a single system and the hybrid block scanners focusing on several ports on multiple systems [3]. In order to perform the previous three scan types attackers commonly use four scan techniques [1] [4]:

Full TCP scan The attacker establishes a TCP connection on a specific port of the target system by completing the TCP handshake. The victim responds with an RST, a SYN-ACK packet or nothing at all, depending on whether the targets port is closed, open, or the host is offline [5].

SYN Scan Contrary to the Full TCP scan, the attacker does not complete the three-way handshake. When the victim responds to the attackers connection request, the attacker terminates the connection setup with an RST packet.

FIN, NULL, XMAS scan A FIN packet sent to an open port on a server is dropped whereas a FIN packet transmitted to a closed port would result in an RST packet response as required by [5]. Similar behaviour is obtained by sending a packet without any flags (null scan), or a packet with all combinations of FIN, URG and PSH flags (XMAS Tree scan).

RST Scan An RST packet arriving at a host is usually dropped, indicating that the host system is (probably) up and running. Otherwise an ICMP host unreachable message is returned.

Additionally attackers usually combine the aforementioned attack types with evasive techniques to prevent detection by NSDS. In this perspective, fragmentation, randomization and slow scanning are frequently mentioned. For a detailed overview of these techniques, we refer to [6],[7],[8],[4].

2.2 Conventional detection approaches

Generally, literature distinguishes between two categories of approaches used by present-day NSDS. The first category uses approaches that rely on flagging N events within a time interval of T seconds. Historically, most NSDS have used detection schemes which can be assigned to this group. The first such algorithm described in literature and implemented by the Network Security Monitor (NSM), used rules to detect a source IP address connecting to more than 15 distinct IP addresses in a small time window [9]. A similar approach is adapted by the lightweight NSDS SNORT, proposed in 1999, which in addition also investigates the contents and form of individual packets in order to detect forged packets [10]. Contrary to SNORT, the NSDS BRO, originating from the same year, utilizes a different technique, which takes the amount of failed network connection attempts into consideration [11]. Further differentiation is possible by extracting application level information from the network packets [11] and by comparing the connection behaviour with expected application patterns [1].

Since both NSM, SNORT and BRO depend on a hard to tune threshold, a second group of detection mechanisms is designed on probabilistic models. The model either derives the probability that a connection is set up, e.g. the chance that a connection to a specific server and port is made [12], or the model estimates the likelihood of the correlation of certain events by taking both the connection destinations, the amount of probes and success ratios into consideration [13],[14].

[1]This research is limited to scan attacks using TCP connections only.

3. SCAN DETECTION APPROACHES USING NETFLOW DATA

In comparison with the conventional approaches described in chapter 2, detection approaches for high speed networks using netflow data constitute a changed environment. Section 3.1 offers a short description of the input data available here, whereas the detection approaches to be developed for them, will be treated in the remainder of this chapter. Subsequently, section 3.2 provides an overview of the detection approaches and the ideas behind them, whereas section 3.3 works them out in more detail.

3.1 Introduction to netflow data

The Netflow protocol has been designed to collect statistical data about the flow of IP packets as they travel across a network interface. The protocol provides basic insights in the network behaviour by grouping together cohesive network packets in a single flow [15]. Successful communication between two hosts, results in two distinct flows: one flow comprises the packets originating from host A to host B, whereas the second flow comprises the packets flowing in the opposite direction. Each flow contains information about the origin (source IP, port), destination (destination IP, port), the layer 3 protocol used, the total amount of exchanged octets and packets, an accumulation of the TCP-flags and flow timestamps (start and end time). For a full overview of the flow record fields, we refer to [16].

3.2 Formulation of generalized detection approaches

As stated generally in 1, netflow data imposes certain restrictions on the detection approaches. Neither is it possible to inspect the payload of individual packets as suggested by [10], nor can one use the approach proposed in [1] to verify consecutive packets against an expected sequence, or check the individual packets semantically. Yet, the attack type and detection techniques summarized in chapter 2 still provide a valuable basis on which to proceed by generalizing and aggregating common characteristics of attack techniques or detection approaches. Using the insights previously stated there, we first formulate three hypotheses, consecutively based on the number of connection attempts, the amount of data packets exchanged during a single connection attempt and,

finally, the connection succes ratio. A fourth new hypothesis will additionally be presented:

1. A vertical scanner is likely to visit more ports on a target computer, than a genuine client would do. Therefore, to detect a vertical scanner, the number of connection attempts to distinct ports has to be taken into consideration. A similar assumption can be adopted for range scanners.
2. Since the attacker's sole interest is information about reachable systems or services running on a specific system, the amount of data exchange will be limited. Complementary, the attack techniques from chapter 2 suggest that the attacker will at most complete the TCP handshake (Full TCP scan) and then close the connection, a process which requires exactly 3 IP packets originating from the attacker. In general, the amount of packets sent from an attacker to a victim will be about 3.
3. A genuine user usually connects directly to the required service but occasionally the service may not exist or the system may not respond. The genuine user will show a constant inbound versus outbound connection success ratio; the number of connections made to a server (outbound) are close to equal to the amount of successful responses coming from the server (inbound). The attacker on the other hand has hardly any knowledge about the target network and will therefore have a deviant inbound-outbound ratio compared to a genuine user; the amount of outbound connections will be significantly larger than the number of inbound connections.
4. Next to the quantity of ports, we assume that the attacker collects information systematically. Consequently, he will not target random IP-port combinations, but he will presumably search an entire port range or IP space. In case of a horizontal or vertical scan, a linear sequence will be revealed, whereas a block scanner clearly shows a block pattern in its connection behaviour. This hypothesis forms an extension to the conventional detection approaches.

3.3 Further elaboration of the detection approaches

A straight approach is only possible for the first of these four assumptions. The other three, on the contrary, require some preparations. In what follows, only vertical scanners will be considered, for reasons of clarity, but it may be obvious that the hypotheses are also valid for the horizontal scanners. In the elaboration of the approaches below, S will be defined as the set of flows originating from host A to a single host B and $S\{field\_name\}$ as a function which returns a set containing the $field\_name$ values from all the flows in S.

3.3.1 Number of connection attempts

The first assumption is derived directly from the netflow data by summing the amount of flows from one source to distinct ports on a single destination.

3.3.2 Limited data exchange

The second hypothesis, focusing on the amount of exchanged data, examines the average amount of packets sent from one host to another. Without further prove we state that the populations in $S\{packets\}$ is not uniformly distributed (in fact a poison distribution is expected). As [17] explains, one should not take the mean value from a non-symmetrical population to calculate an average. Instead the median function should be used for skewed populations. However, intuitively the median value of for instance $S\{packets\} = \{1000, 1100, 1200, 50000, 60000\} = 1200$ is a bad indicative for an average value since 40% of the values are above or equal to 50000. To overcome this problem, a subset of $S\{packets\}$ will be taken, hereby omitting outlier values from the original subset [18]:

1. Select a subset Soctets
   $S'\{packets\} = \{\forall x \in S'\{packets\} \mid [LOW, HIGH]\}$
   LOW and $HIGH$ are determined by deriving a boxplot from $S\{packets\}$ [18].
2. Calculate the mean over the elements in $S'\{packets\}$.

For the earlier mentioned example, the newly calculated average would amount to 22660.

3.3.3 Connection Fail Ratio

The third hypothesis measures the amount of inbound and outbound connections to calculate the connection success ratio, or opposite, the connection failure ratio (CFR).

$$CFR = \frac{outbound\ connections - inbound\ responses}{outbound\ connections} \quad (1)$$

As expected, genuine users will have a CFR ratio of roughly 0. [19] provides further insight in the decision criteria used to flag scanners based on their CFR. Depending on the false positive and false negative requirements, 3 to 24 connection attempts have to be taken into consideration and a CFR of about 0.5 to make an accurate decision [19]. The attacker can try to influence its CFR by regularly alternating between connecting to open ports, which are discovered in earlier scan attempts, and connecting to not yet examined ports. To overcome this problem, only unique connection attempts will be taken into consideration. In this perspective, a unique connection is the first occurrence (in time) of a connection from host A to host B on port C.

3.3.4 Linear pattern in connection behaviour

Vertical scanners presumably probe several ports on a target system in a systematical fashion. To determine such a linear relationship between a list of numbers, in this case port numbers, a statistical method called Linear Regression may be used. This method models the relationship between dependent variables Y, the port numbers, and the independent variables $X_i = i$ where i = the i^{th} member of the set[2] $S\{dst\_port\}$. The model determines the best-fit line through the observed ports, thereby calculating the slope (A) and the Y intercept (B). The line, described by the formula $\Psi i = A * X_i + B$, is estimated by minimizing $\sum_i (\delta_i)^2$ where δ is the vertical distance between the line Ψ and the actual port (Figure 2). For a detailed discussion on Linear Regression, we refer to [20].

Finally, Pearsons coefficient of regression [20], R^2, is used to approximate how well the line corresponds to the port numbers. R^2 is a value between 0 on the one hand, indicating that none of the points fit the regression line, and 1 on the other hand, indicating the opposite.

[2]The elements in this set have to be ordered in an ascending order to work for the Regression algorithm.

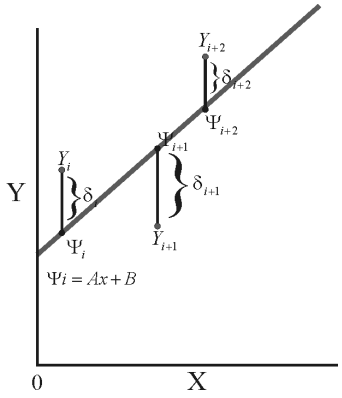

Figure 2: Linear regression

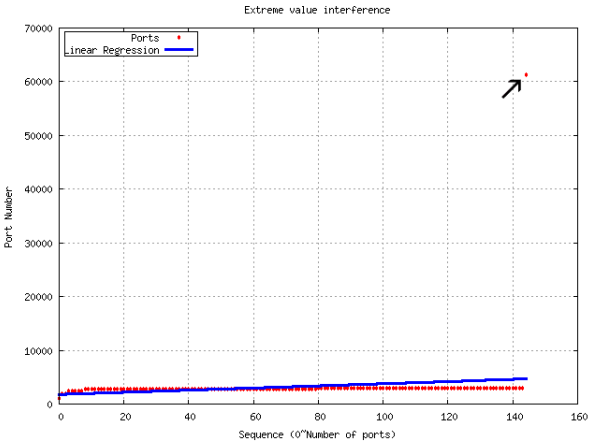

Figure 3: Influence by extreme value

In our discussing R^2 indicates how well the observed connection behaviour maintains a linear correlation. In this perspective we anticipate that the observed connections of a genuine host, contrary to scanners, have a very weak linear relation at most.

One minor note should be taken: the suggested approach is heavily affected by extreme values (Figure 3).

Again, the boxplot function is used to remove the outlier values from $S\{port\_dst\}$, prior to the calculation of the regression line, resulting in a much better estimation of the line. The light grey line in (Figure 4) nearly intersects all the points whereas the dark grey line is identical to the line in (Figure 3).

3.3.5 Block pattern detection in connection behaviour

In the following, $S2$ is defined as the set of flows originating from host A. By way of illustration, Table 1 is constructed from a fictious $S2\{(ipv4\_dst, port\_dst)\}$.

Table 1 reveals two equally sized block patterns consisting of 6 connections each: 10.0.0.{1,2,3}*{1,4} and 10.0.0.{1,2}*{1,2,4}. Table 1 is represented in a bipartite graph [21] where set R contains the IPs, set C the ports and every edge in the graph is a single connection (Figure 5). The two block patterns, forming bicliques [22] in the bipartite graph, are depicted with bold lines. Finding the size of the largest block pattern in $S2$ is now rephrased to finding the maximum edge biclique in a bipartite graph. An estimation to this NP-complete search problem [23] is presented by [24].

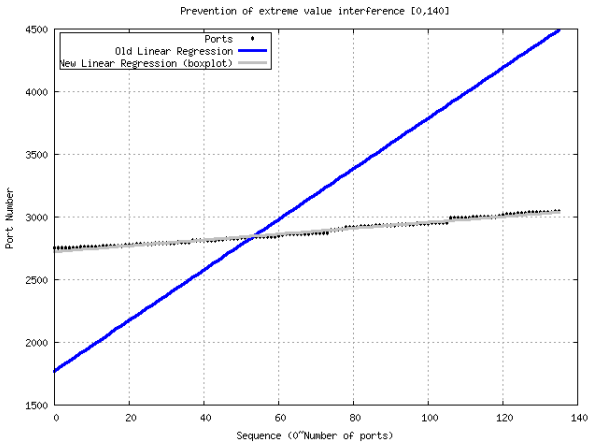

Figure 4: Removal of influence by extreme value

Table 1: Fictious set of connections originating from a single host

Dest. IP \ Dest. Port	1	2	3	4	5
10.0.0.1	*	*		*	
10.0.0.2	*	*		*	
10.0.0.3	*		*	*	
10.0.0.4			*		*

Since we are interested in finding block patterns with a minimal size of 2*2, some preprocessing of the data is required to prevent the algorithm of returning patterns with a $Y * 1$ dimension. This problem is efficiently solved by removing connections that are not part of a minimal 2*2 block, like the connections 10.0.0.4:{3,5} and 10.0.0.3:3.

4. APPLICATION OF THE DETECTION APPROACHES

The detection approaches elaborated in chapter 3 can now be put into practice. First, in section 4.1, an overview of the test environment is given, providing information about the data used for

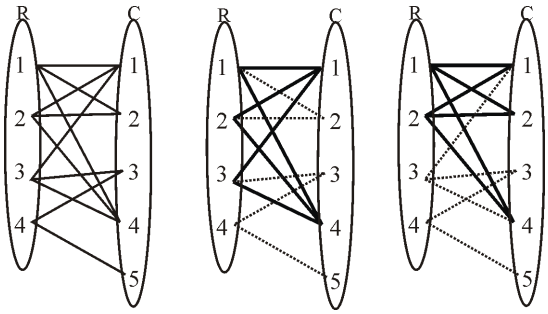

Figure 5: DestIP and DestPort in bipartite graph

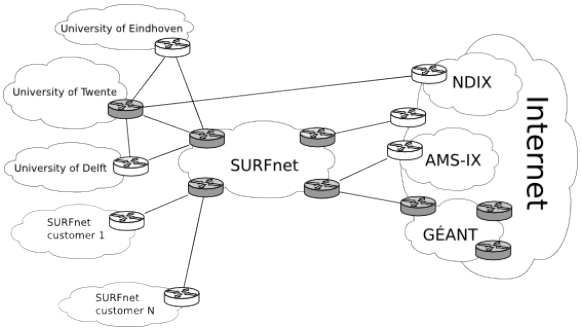

Figure 6: Test environment

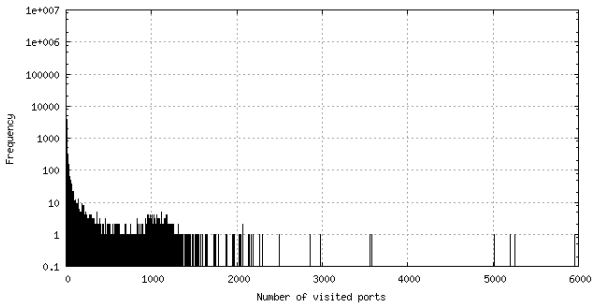

Figure 7: Frequency diagram of visited ports

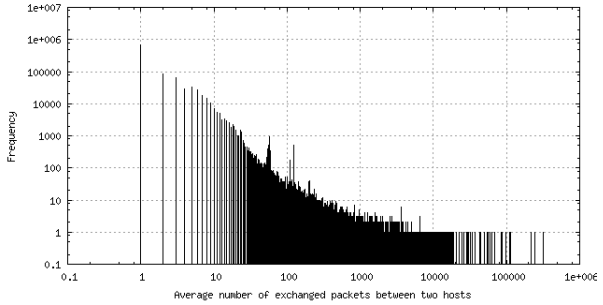

Figure 8: Frequency diagram of exchanged packets

the analysis of the approaches. Second, the results obtained by the detection approaches are presented in section 4.2.1 through 4.2.5. Finally, using a combination of the detection approaches, a selection of potential scanners is created, which is validated using two different reference sets in section 4.3.

4.1 Test environment

To allow for a thorough analysis of the proposed approaches in chapter 3, one week of netflow records were recorded, starting at 31 July 2007. The netflow collector used, was located at the entrance of the class B network of the University of Twente (Figure 6). From the week of netflow data, which occupies more than 200GB, 2 days were taken and stored in a database. Finally, for this research, a three hour subset (1 Aug 2007 01:00:00 GMT-1 Aug 2007 04:00:00 GMT), comprising 23.5 million netflow records[3], was created. The server hardware and the computational time required to execute the detection approaches, restricted us from taking a wider time span of netflow data.
The approaches elaborated in chapter 3.3 are implemented in the SQL database language. In addition several Java programs were written to optimize parts of the SQL statements, thereby reducing the required execution time considerably.

4.2 Results of the detection approaches

In the following subsections, the approaches elaborated in section 3.2 are applied on the subset of netflow data. For each approach, the results are summarized.

4.2.1 Number of connection attempts

The first frequency diagram (Figure 7) depicts the number of visited ports by a single client on a single host. Considering just over 1 million distinct <source,destination> pairs, the average amount of visited ports is 2.1. Almost 93% of the participating clients visit just 1 distinct port on a server. Although the graph shows a rapidly decreasing curve, it is impossible to identify a clear border between genuine traffic and anomalous scan traffic solely based on the amount of visited ports

4.2.2 Limited data exchange

(Figure 8) provides insight in the amounts of data send from the client to the server. On average 10.9 packets are sent. Based on the attack techniques described in section 2.1, we hypothesized that the data exchange between a victim and a scanner is very limited. In fact, to perform the attack techniques described in chapter 2.1, at most 3 data packets are required (Full TCP Scan). Unfortunately in no less than 80% of the cases, the netflow collector observed a transmission of 3 packets or less from a client to a server, as (Figure 8) reveals. This means that filtering on the amount of transmitted packets does not provide a clear distinction between scanners and genuine clients.

4.2.3 Connection fail ratio

(Figure 9) provides insight in the occurrence frequency of Connection Fail Ratios (CFR). A Connection fail ratio of 1 implies that all connections initiated by a single client to a single server failed, e.g. no response was returned by the server. As expected, two peaks at a CFR of 0 and a CFR of 1 are visible. Interesting to note is the strange peak at a CFR of 0.5, indicating that exactly half the connections were successful. Overall, with 0.72, the average CFR appears to be quite high.
(Figure 10) plots the CFR against the number of distinctly visited ports on a single server. The first hypothesis assumed that scanners initiate a more than average amount of distinct connections to a single system, whereas the third hypothesis expects a high CFR rate for scanners. Contradictory to the expectations of the combination of hypothesis 1 and 3, (Figure 10) reveals no clear relation between them: an increase in the number of visited ports does not cause an increase in the CFR.

4.2.4 Linear connection pattern in visited ports

(Figure 11) depicts how strong the linear relation is between the ports connected to by a client on a single system. An R^2 of 1 implies a full linear relation whereas an R^2 of 0 means the opposite. Similar to (Figure 10), (Figure 11) plots the R^2 against the number of distinctly visited ports. This graph makes clear that the larger the amount of visited ports, the stronger the linear relation

[3]This includes TCP traffic only.

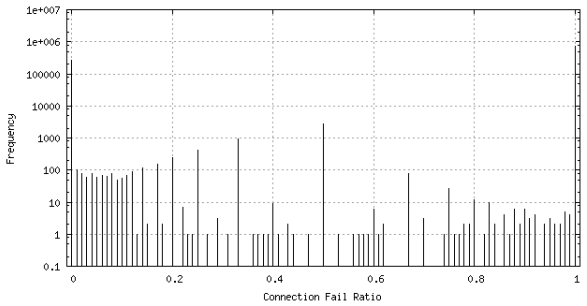

Figure 9: Frequency diagram of Connection Fail Ratio

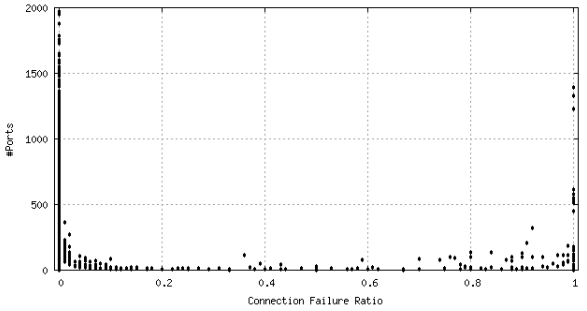

Figure 10: Distribution of CFR over the number of visited ports

becomes. This appears to be consistent with the expectations formulated in chapter 3.3.1 and 3.3.4, stating that a scanner both has a large number of distinct connections as well as a strong linear relation in its behaviour.
Finally, one may notice that most of <source,destination> pairs have a large R^2 (>0.7). This negative side effect is caused by the boxplot method [chapter 3.3.4] used to prevent the interference of extreme values as depicted in (Figure 3). Since almost all the <source,destination> pairs have a fairly large R^2 (average of 0.95[4]), it is impossible to distinguish vertical scanners from genuine clients based on their connection pattern.

[4]This average is calculated over for all the clients who visited more than 1 distinct port on a target system. After all there exists no such thing as a linear relation between just 1 port.

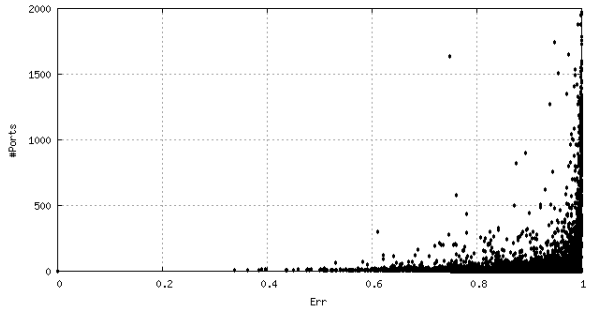

Figure 11: Distribution of R^2 over the number of visited ports

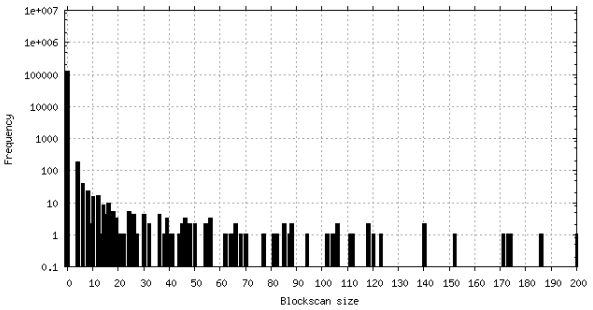

Figure 12: Frequency diagram of block scan size

4.2.5 Block connection pattern

The last approach suggested in chapter 3 focused on the detection of block patterns in the connection behaviour of a source. (Figure 12) reveals the strong differentiating character of this approach. Of the 117870 distinct sources in our 3 hour subset, 117489 do not have a block pattern. After removal of the sources with a block size smaller than 10, which cannot be considered as real block scanners, only 109 sources remain which can be further analyzed with the other detection approaches from chapter 3. Interesting to mention is that most of the larger blocks detected have a thin, rectangular shape. Only on some very rare occasions outside the here presented test set of netflow data, large, nearly square sizes of block patterns occurred. This complies with the findings of [3].

4.3 Final validation

The first approach of the number of connection attempts provides us with a valuable means to considerably reduce the number of sources in the set of netflow records. It excludes those combinations which cannot be seriously considered as scanners, due to their low number of distinct connections.
In combination with the second approach the subset thus obtained, can be further reduced by removing the records indicating an extensive data exchange. Since there appears to be no linear relation between the number of visited ports and the CFR, as discussed in 4.2.3, filtering on the CFR would even likely remove potential scanners. Therefore the CFR-approach is not used to further decrease the subset. The same applies to the fourth approach, as all observed patterns maintain a fairly high R^2. Consequently it denies us the possibility to distinguish between genuine and malicious traffic. The block pattern detection approach, on the contrary, will prove to be useful at a later stage of our validation.
Application of the first and second approach to our original subset leaves us with a very small set of netflow records, which we may assume to contain a high density of scanners. In this perspective, all the <source,destination> pairs were removed where the average transmitted amount of packets was > 20 and the amount of distinctly visited ports <20. This reduced the original number of <source,destination> pairs from 1 million to 4713. This filtered subset is used during the validation process.

Contrary to experiments in a closed setting, a problem with an authentic set of netflow data is the lack of a reference set which can be used to verify the achieved results. To overcome this problem, we first thought that the TCPDump header files, generated complementary to the collection of the netflow data, could provide

further insight. Allthough these header files provide more details about the order in which packets were exchanged, in many cases the information is not sufficient for a clear validation.
As a solution, two new validation sets are created, each of which contains traffic which can be attributed to scan activities with a high degree of probability. The following two validation sets will next be compared with the contents of the filtered subset.

1. All the records of the sources which transmit on average just a single packet. As a TCP-handshake requires at least three packets for a correct termination of a connection, the sending of an average of 1 packet can be considered as highly remarkable, if not suspicious.

2. A block scan, which may be conceived as a vertical and a horizontal scan performed simultaneously (Figure 1). The block scan detection approach is capable of distinguishing between sources which do not reveal a block pattern in their connection behaviour (size 0) which make up the absolute majority, on the one hand, and, on the other, a very limited group which does reveal a block pattern. Of the latter group those with the largest block patterns are in all likelihood block scanners. Therefore the top 20 results of the block scan detection will be compared with the records in our filtered subset.

The first comparison set contains 657509 unique combinations of source and destination. After removal of the range scanners 55224 combinations are left. Range scanners are assumed to connect at least 20 or more times to distinct destinations on the same port. Of the 55224 cases of suspiscious <source,destination> pairs left, 334 appeared to occur in our own filtered set of port scanners, whereas 54881 did not. This is mainly due to a number of advanced, distributed scans which the algorithms failed to recognize. Their exact number could not be determined. Also cases where the amount of visited distinct ports were less than 20 occurred, but these were too small to be positively identified as scanners.

The second comparison set includes the results generated by the block scan detection approach. A selection of the first ten results will do to demonstrate our point. The table in Appendix A illustrates that the detected block scans were also identified as port scanners. In most cases the scanner executed a port scan over a large number of ports on two or three systems. Due to the overlap in the port numbers these port scans were also identified using the block scan detection approach.

Finally, the detecion approaches based on the linear pattern detection and the CFR were validated. As already expected by the results provided in section 4.2.3 and 4.2.4 these two approaches are not suitable for detection. In our validation sets, the R^2 coefficient, indicating the power of the linear relation, fluctuated between 0.65 and 1. The same goes for the CFR; in 6% of the cases, in our validation sets, the CFR equals 0 (see also Appendix A). To summarize the results, all cases our two validation methods identified with reasonable certainty as scanners, were included in the filtered subset[5]. After substraction of the number of validated hosts from the number of hosts in our filtered subset, 4359 were not verifiable with the information provided by both netflow data and the TCP header files. Nevertheless, the original list of 117870 sources was reduced to 1128 suspicious hosts. Packet filtering rules may be written to redirect traffic from these hosts to traditional packed based NSDSs, for a more detailed inspection.

[5]This excludes the occurrences of the distributed scanners.

5. CONCLUSION

This research has attempted to demonstrate the possibility of detecting scan attacks in netflow data traffic. First four scan techniques were discussed [section 2.1], as well as the detection systems usually deployed against them [section 2.2]. The transition from low to high speed networks required the development of new detection approaches. From the conventional packet-based detection systems a set of general detection approaches has been derived [section 3]. In addition a new approach has been developed focussing on patterns in the connection behaviour for both linear and block scanners [section 3.3.5]. Their application on netflow data demonstrates that positive results can be achieved in filtering suspected traffic. Our approaches have been applied to 23 million authentic netflow records. Furthermore, validation was performed by comparison with 2 sets of records, consisting of highly suspicious hosts. This validation confirmed the presence of scan attempts [section 4].
Nevertheless the subject matter is definitely complex. For instance, taking into account the limited amount of information available in a netflow record, it will inevitably often remain unclear whether a scan attempt is taking place. Even if identification of every netflow record as belonging to genuine or malicious behaviour is not yet feasible, research, as in this paper, may provide a gentle step in the right direction by improving filtering capabilities. Packets belonging to the reduced and much smaller set of suspicious sources obtained by our detection approaches, may be forwarded for further analysis to a more fine grained detection mechanism, capable of packet level inspection.

REFERENCES

[1] R. Sekar, Y. Guang, S. Verma, and T. Shanbhag, "A high-performance network intrusion detection system," in *CCS '99: Proceedings of the 6th ACM conference on Computer and communications security*. New York, NY, USA: ACM, 1999, pp. 8–17.

[2] R. R. Kompella, S. Singh, and G. Varghese, "On scalable attack detection in the network," in *IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*. New York, NY, USA: ACM, 2004, pp. 187–200.

[3] C. S. E. Lee, C.B. Roedel, "Detection and characterization of port scan attacks," Ph.D. dissertation, University of California, San Diego, 2003.

[4] M. de Vivo, E. Carrasco, G. Isern, and G. O. de Vivo, "A review of port scanning techniques," *SIGCOMM Comput. Commun. Rev.*, vol. 29, no. 2, pp. 41–48, 1999.

[5] "Rfc 793. transmission control protocol, darpa internet program, protocol specification," 1981.

[6] "Nmap - free security scanner for network exploration and security audits." [Online]. Available: http://nmap.org/

[7] J. Green, D. Marchette, S. Northcutt, and B. Ralph, "Analysis techniques for detecting coordinated attacks and probes," in *ID'99: Proceedings of the 1st conference on Workshop on Intrusion Detection and Network Monitoring*. Berkeley, CA, USA: USENIX Association, 1999, pp. 1–1.

[8] M. G. Kang, J. Caballero, and D. Song, *Detection of Intrusions and Malware, and Vulnerability Assessment*, 2007, ch. Distributed Evasive Scan Techniques and Countermeasures, pp. 157 – 174.

[9] L. Heberlein, G. Dias, K. Levitt, B. Mukherjee, J. Wood, and D. Wolber, "A network security monitor," *Research in Security and Privacy, 1990. Proceedings., 1990 IEEE Computer Society Symposium on*, pp. 296–304, May 1990.

[10] M. Roesch, "Snort - lightweight intrusion detection for networks," in *LISA '99: Proceedings of the 13th USENIX conference on System administration*. Berkeley, CA, USA: USENIX Association, 1999, pp. 229–238.

[11] V. Paxson, "Bro: a system for detecting network intruders in real-time," in *SSYM'98: Proceedings of the 7th conference on USENIX Security Symposium, 1998*. Berkeley, CA, USA: USENIX Association, 1998, pp. 3–3.

[12] C. Leckie and R. Kotagiri, "A probabilistic approach to detecting network scans," *Network Operations and Management Symposium, 2002. NOMS 2002. 2002 IEEE/IFIP*, pp. 359–372, 2002.

[13] S. Staniford, J. A. Hoagland, and J. M. McAlerney, "Practical automated detection of stealthy portscans," *J. Comput. Secur.*, vol. 10, no. 1-2, pp. 105–136, 2002.

[14] J. Jung, V. Paxson, A. Berger, and H. Balakrishnan, "Fast portscan detection using sequential hypothesis testing," *Security and Privacy, 2004. Proceedings. 2004 IEEE Symposium on*, pp. 211–225, May 2004.

[15] "Introduction to cisco ios netflow." Cisco white papers. [Online]. Available: http://www.cisco.com

[16] "Nefflow export datagram format." [Online]. Available: http://www.cisco.com/en/US/docs/net_mgmt/netflow_collection_engine/3.0/user/guide/nfcform.html

[17] "Mean or median." [Online]. Available: http://www.conceptstew.co.uk/PAGES/mean_or_median.html

[18] "Boxplot." [Online]. Available: http://www.corda.com/docsource/doc7/Manuals/graph_ref/box_plot_graphs.htm

[19] H. Nam, S. Kim, "Scanner detection based on connection attempt success ratio with guaranteed false positive and false negative probabilitie." Ph.D. dissertation, CyLab, Carnegie Mellon University, 2006.

[20] "Linear regression." [Online]. Available: http://en.wikipedia.org/wiki/Linear_regression

[21] "Bipartite graph." [Online]. Available: http://en.wikipedia.org/wiki/Bipartite_graph

[22] "Complete bipartite graph." [Online]. Available: http://en.wikipedia.org/wiki/Complete_bipartite_graph

[23] R. Peeters, "The maximum edge biclique problem is np-complete," *Discrete Appl. Math.*, vol. 131, no. 3, pp. 651–654, 2003.

[24] C. Ding, Y. Zhang, T. Li, and S. R. Holbrook, "Biclustering protein complex interactions with a biclique finding algorithm," in *ICDM '06: Proceedings of the Sixth International Conference on Data Mining*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 178–187.

APPENDIX A: RESULTS OF THE BLOCK-DETECTION APPROACH

Table 2: caption Table 1

Block Detection:				Information in filtered subset:			
Nr	Block Size	IPs in Block	Ports in Block	Ports (max over participating IPs)	CFR (Average)	R2 (Average)	Packets (Average)
1	990	9	110	113	0.6	0.95	1
2	422	2	211	2265	0	0.92	17
3	172	2	172	3563	0	0.99	5
4	332	2	166	5955	0	0.99	17
5	330	3	110	966	0	0.93	11
6	322	2	161	177	0	0.99	8
7	288	2	144	1394	0.99	0.99	12
8	186	2	93	1210	0	0.99	5
9	171	3	57	800	0.60	0.98	8
10	120	2	60	180	0	0.99	1