PORTSCAN DETECTION USING NETFLOW DATA

Matěj Grégr
Doctoral Degree Programme (1), FIT BUT
E-mail: igregr@fit.vutbr.cz

Supervised by: Miroslav Švéda
E-mail: sveda@fit.vutbr.cz

ABSTRACT

Portscan detection methods are usually focused on enterprise networks where the traffic volume is low. Portscan detection on high speed backbone networks has however different requirements. This paper introduces a method for detection of portscans on a university backbone network using NetFlow data, collected by hardware accelerated NetFlow probes.

1 INTRODUCTION

To realize a successful attack in a network, the attacker needs to know which computers in the network are accessible and which services that computers provide. For network mapping it can be used techniques, such as scanning of IP address, ports, eavesdropping or querying services on application layer, e.g. DNS.

For the detection of portscan activity and other network attacks, Network Intrusion Detection Systems (NIDS) are used. NIDS systems use sensors placed on the edge of the network. Sensors provide logs or other types of data which are analyzed by NIDS. Two main approaches are used, network pattern recognition and anomaly detection. In network pattern recognition, attacks are revealed by matching a network traffic with some known network pattern. Anomaly based detection detects attacks using deviation from the standard and legit network traffic.

Logs from syslog, data collected from network protocols such as SNMP, NetFlow, sFlow or fully recorded network communication with `tcpdump` program are possible inputs to NIDS system. This paper focuses on analysis of NetFlow data and introduces a method for detecting portscan with NetFlow data.

2 RELATED WORK

NIDS systems use different techniques for portscan detection. In this section, some often used techniques, presented in literature, are described. There has been many more techniques used by commercial systems but their approach and methods have not been published.

NIDS system `Snort` is one of the first systems where portscan detection algorithm was introduced. `Snort` detects portscans when fast repeat attempts to connect to a service or an IP address occur. System also controls packets if they are not adjusted for different types of scanning such as NULL or XMAS scanning. The main disadvantage of this is that the attacker can easily circumvent this detection method by insert time delay between packets. False positive or negative ratio highly depend upon correct determination of timers.

More advanced method is Threshold Random Walk (TRW) [2]. Idea of this method is based on the fact that the main characteristics of scanners is that they are more likely to choose hosts that do not exist or do not have the requested services available. The reason stems from the lack of knowledge of which hosts and ports on the target network are currently active. TRW for every IP address computes the number of events *e*. If the connection was successful, *e* is decreased and if the connection was unsuccessful the value is increased. As far as the number of events overruns a threshold, the IP address is marked as being used by a scanner. This method is implemented in NIDS system `Bro`.

Other NIDS systems such as MINDS [4] or ADAM use network anomaly based detection. ADAM uses association rules and apriori algorithm while the MINDS system detects network attack using Local Outlier Factor method. These systems are quite obsolete. Recent NIDS system is for example CAMNEP [1]. This system uses several known published methods, which are used for determination of the confidentiality of network flow. It is also one of the few systems which use NetFlow data and can be deployed on high-speed networks.

3 BRNO UNIVERSITY OF TECHNOLOGY BACKBONE STATISTICS

NetFlow data are generated by routers or hardware accelerated probes. The amount of data depends on the network traffic and can vary from several MBs in small networks to several hundreds of MBs in very large networks.

For analysis and scan detection, data from the Brno University of Technology (BUT) backbone network are used. These data are collected from two 10 Gb/s lines which connect BUT network to CESNET academy network. Techniques used for scan detection are usually deployed on enterprise networks where the traffic volume is low. Several statistics were created to verify if these techniques can be used also on backbone links. The BUT network has 3300 flows/sec average. Figure 1 shows the amount of unique IP addresses during the day. Average is around 2 million of unique IP addresses during peak hours. Figure 2 shows how many packets are usually contained in a flow. It can be seen that flows with only one packet are the most frequented. In fact, 55% of flows are flows with just one packet and flows up to 3 packets present approximately 80% of all flows. This put the scan detection much more difficult because scan tools creates mainly one packet flows when scanning some network host.

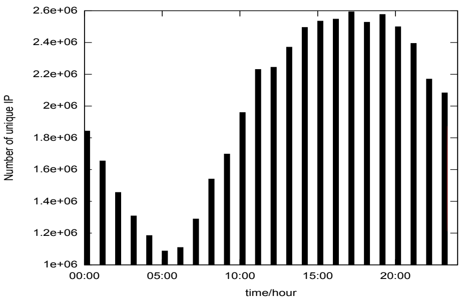

Figure 1: Number of unique IP address

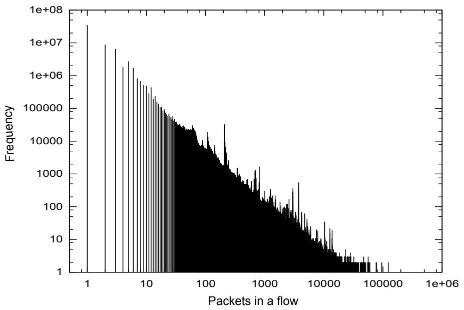

Figure 2: Number of packets in a flow

A free security network scanner `nmap` in its default configuration scans a thousand of the most frequently used ports. As Figure 3 shows a benign user usually tries to connect to small number of ports. In 80% cases it is just one port. On Figure 4, it can be seen that the number of IP addresses created just by one flow is the greatest one but it forms only 6% of the total quantity. Most IP addresses create several flows.

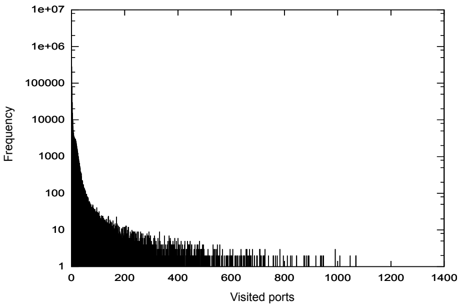

Figure 3: Number of visited ports

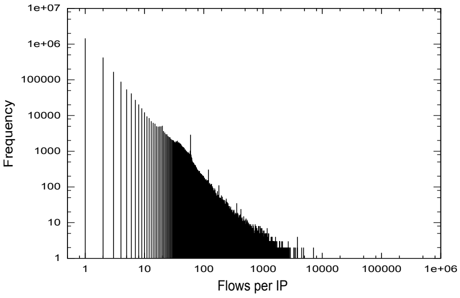

Figure 4: Number of flows per IP address

All presented graphs were created from NetFlow records collected during one month. The amount of data analyzed for presented graphs is 215,6 GB of compressed NetFlow records, which corresponds to more or less 650 GB of uncompressed NetFlow data.

4 PORTSCAN DETECTION

From obtained statistics it is obvious that the methods for detection of scanning used by NIDS systems snort or Bro have a problem with large amount of IP addresses and connections. These methods also do not work with NetFlow data but monitor network connections in real time. On the other hand, NetFlow data can be analyzed after they has been exported from a probe to a collector (usually within five minutes).

It is very difficult to detect all scans in reasonable time in large networks containing millions of connections. If an attacker scans just several ports on an active host, this scanning would not be visible and detectable in several hundreds of NetFlow records that are stored on backbone links. It is not our purpose to detect all scanings in very large network; the goal arises to detect attackers that horizontally scan whole subnets or perform big block scans. Detecting this kind of scanning in time is desirable because computers infected with worms usually act like that, and network administrators need to detect these computers as fast as they can, before the computers harm the network services.

The analysis of how scanning using `nmap` tool can be detected in NetFlow records was performed. When `nmap` scans a port it creates one flow with only SYN flag. If it scans one thousand of ports, one thousand of one packet flows are created. Single packets also contain the same number of bytes. To reduce the number of IP addresses to analyze, the top N statistics sorted by number of flows is created. These statistics assure that only IP addresses with many flows would be analyzed. Because an attacker creates many flows, attacker's IP address should appear in the statistics.

4.1 DETECTION USING DECISION TREES

Portscan can be detected using decision trees. We created this method mainly because we did not have completed NetFlow records from BUT network. NetFlow records were created using software probe which did not save TCP flags, ICMP types, ICMP codes and other fields. From that NetFlow records it can not be determined if the flow of one packet is a start of scanning or just a one-packet answer of some network protocol. Using algorithm ID3 [3] with entropy used for determining which item is the most important from a training set, the decision tree is created. Training set contains observed values - destination IP address, destination port, number of packets and bytes in a flow and determination if the flow is suspicious from scanning point of view or not. Example of the line in a training set is following: flow, where an IP address is the same as the IP address of the previous flow, the difference between destination ports is small, number of bytes is same and number of packets is less than three. This can mark a flow as a vertical scan. Small differences in destination ports mean the ports go sequentially. For example a flow contains destination port 110 and the following flow contains destination port 111. It is the way how vertical scan is made.

From the previous step where we created the top N statistics, we obtained IP addresses with the highest number of flows. Traffic, where the source IP address is the address from the statistics is then filtered out of the NetFlow record. The decision tree is transformed into a set of `if-then-else` rules and with these rules the filtered traffic is analyzed. For every IP address (similarly to TRW method) the number of events *e* is computed. If the tree marks the flow as scanning, the value *e* is increased. If *e* overruns selected threshold, the IP address is marked as an attacker.

4.2 DECISION TREES WITH TCP FLAG FILTRATION

Recently, hardware accelerated NetFlow probes have been installed into the BUT network. These probes collect complete NetFlow records. When the top N statistics of top talkers is created, it usually contains some highly loaded servers which create a plenty of flows. With complete NetFlow records, it is possible to generate statistics just with IP addresses that contain a lot of flows with SYN or RST+ACK flags. This filtering decreases the number of addresses needed to be analyzed. The traffic between obtained addresses is filtered out and sent as the input into the decision tree as it was described in section 4.1. The decision tree then decides if the IP address is an attacker or not. The reason why we need this decision and can not simply mark the IP address with high number of SYN flows as an attacker is that the IP address can be a client trying to connect to a server, which is temporaly inaccessible.

With the filtering presented, it is possible to detect only TCP and UDP protocols. However, it is not a problem to add another conditions which allow ICMP messages type 8 and 0 (Echo Request and Echo Reply). After that it is possible to detect scans with the same successful rate as using decision trees. In addition, the detection time is decreased because the top N statistics does not contain servers with a lot of flows but benign traffic.

5 EVALUATION

Presented methods were implemented and tested using real NetFlow data from the backbone BUT network. Graphs in Figure 5 show a lot of detected scans. The network administrator then need to decide how this information can be used to increase security of the network.

First thing which comes to mind, is to block detected IP addresses. In a large network this solution can bring more problems than it solves.

- The time of the blocking is very hard to determine. Short blocking will not be very successful while long time blocking creates a lot of rules for the network firewall.

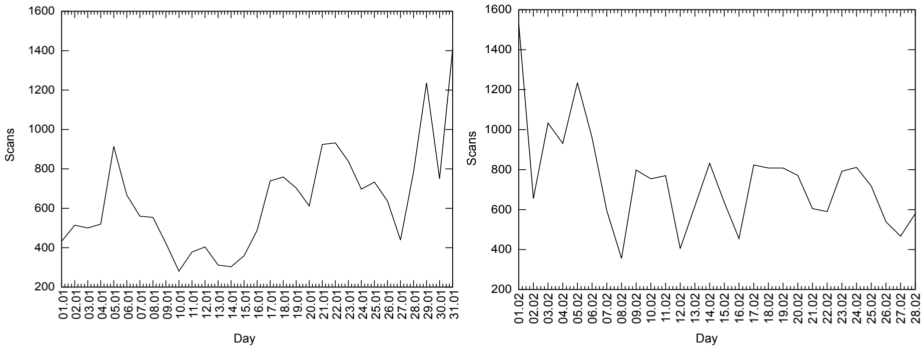

Figure 5: Number of detected scans

- Scanning tools can be easily set to use forge IP address. After that, legitimate access to the network can be blocked.
- An attacker can be hidden behind a device providing Network Address Translation. Blocking of such IP address can cause blocking of other ligitimate users.

Considering these limitations, the scan detection methods were updated to meet the needs of the BUT network. Methods detect every scan but only scans originating outside the BUT network are sent to the administrator. The reason for this is that for computers in the BUT network an administrator who supervises the computer can be found. The administrator then can assure that the computer will be blocked etc.

6 CONCLUSION

This paper presents and analyzes statistics of high-speed university backbone taken from NetFlow records. Based on these statistics we propose a method how to detect portscan attacks in these large high-speed networks. For scan detections, NetFlow records collected using hardware accelerated probes are used. Presented method has been implemented and evaluated. Results are described in section 5. The system now runs on the CVIS server.

This work was partially supported by the BUT FIT grant FIT-S-10-1 and the research plan MSM0021630528.

REFERENCES

[1] Řehák, M., Pěchouček, M., Čelada, P.: CAMNEP: An intrusion detection system for high-speed networks. In: Int. Conference on Autonomous Agents, Estoril, PT, 2008, s. 133-136.

[2] Jung, J., Paxson, V., Berger, A.: Fast Portscan Detection Using Sequential Hypothesis Testing. In: Proc. IEEE Security and Privacy, USA, 2004, s. 211-225, ISSN 1081-6011.

[3] Quinlan, J.: Induction of Decision Trees. Machine Learning., vol. 1, num. 1: p. 81-106, ISSN 0885-6125.

[4] Ertöz, L., Eilertson, E., Lazarevic, A.: The MINDS - Minnesota Intrusion Detection System. MIT Press, 2004.