Analysis of Vertical Scans Discovered by Naive Detection

Tomas Cejka, Marek Svepes

Analysis of Vertical Scans Discovered by Naive Detection

Tomas Cejka[1], Marek Svepes[2]

[1] CESNET, a.l.e., Zikova 4, 160 00 Prague 6, Czech Republic
cejkat@cesnet.cz
[2] CTU in Prague, FIT, Thakurova 9, 160 00 Prague 6, Czech Republic
svepemar@fit.cvut.cz

Abstract. Network scans are very common and frequent events that appear in almost every network. Generally, the scans are quite harmless. Scanning can be useful for network operators, who need to know state of their infrastructures. Contrary, scans can be used also for gathering sensitive information by attackers. This paper describes a simple detection method that was used to detect vertical scans. Our aim is to show results of long-term measurement on backbone network and to show that it is possible to detect scans efficiently even with a simple method. The paper presents several interesting statistics that characterize network behavior and scanning frequency in a large high-speed national academic network.

1 Introduction

Network scanning is a common and frequent activity that can be observed in almost every network infrastructure. It is a normal benign mechanism used by network operators or automatic tools for monitoring and management. A network scan is based on probing targets to recognize the active ones. That is a scan referred as *horizontal*. Scans can also probe ports of one target. Such scans are called *vertical*. A *block* scan is a combination of horizontal and vertical scans.

Scans are easily performed even by attackers. Attackers can use scanning to search for publicly available services and vulnerable devices in the internet. Even though network scanning is basically harmless, current researches show, that it can be dangerous in some ways. Bartos et al. in [2] show correlation between network scans and attacks (e.g. bruteforce guessing of passwords or DoS attacks) that follow scans. Similarly in [11], Raftopoulos et al. discuss their observation about high probability of malware infection of devices that had been scanned previously. Therefore, it is important not to underestimate a danger of scans.

Unfortunately, there is no universal detection method, that would be suitable for all sizes of networks. According to [13], large transit networks or National Research and Education Network (NREN) infrastructures require a special detection approach. The main issues related to such networks are: high speed, wider diversity of IP addresses, lack of knowledge about end-hosts' configuration, asymmetric routing, coexistence with other monitoring and detection tasks without interference. This paper presents observations from the perimeter of

CESNET2. It is Czech NREN, a backbone and a transit network. Based on observations, we created a straightforward detection method.

2 Related Work

Bhuyan et al. presents a taxonomy of network scanning and a survey of some existing detection approaches in [3]. Using the taxonomy, we can classify the detection method presented in our paper as a threshold-based method.

One of the well-known methods is a Threshold Random Walk (TRW) proposed for scan detection by Jung et al. in [7]. The detection method was implemented as a part of Bro [10]. Sridharan et al. in [13] points out disadvantage of TRW that needs knowledge about the configuration of end-hosts. In backbone networks there are several issues that complicate scan detection. However, it is still useful to perform detection even on backbone level. The paper investigates effectiveness of existing methods and proposes a new method Time-based Access Pattern Sequential hypothesis testing (TAPS). Lee et al. is one of the most closely related works to this paper. Their paper presents a report about observed port scans. The authors analyzed two weeks of traffic at University of California, San Diego (UCSD) using Snort [12]. The paper was written in 2003 and the authors discovered 9,927 vertical scans. Whereas, we have been monitoring network traffic from CESNET2 more recently (2015) for longer time (two months) and average number of discovered vertical scans in two weeks was 203,000.

As it was shown, there are various approaches of scan detection. However, we used a simple flow-based method with thresholds and filtering flow records.

3 Detection Algorithm

The detection algorithm uses information from basic flow records (source and destination IP addresses and ports, protocol, #packets, #bytes). The principle of algorithm was inspired by characteristics of scans generated by nmap [8]. Analysis showed that scans are composed of plenty flow records with small number of packets (≤ 4) transferred between the source IP (potential attacker) and a destination IP (victim).

We have focused on a default scanning technique supported by nmap. It uses Transmission Control Protocol (TCP) packets with set SYN flag. This simulates establishment of a new TCP session and the target should reply with SYN+ACK if the probed port is opened. The detection results of SYN scans are verifiable manually even in unknown network traffic of backbone since TCP normal traffic from a host always contain not only SYN flag and should not imply plenty RST responses. Detection of other scan types is more complicated due to verification of false positives and missing ground truth in the real backbone traffic.

The detection algorithm is based on analysis of the number of destination ports per source IP and uses threshold for number of ports. It is important to remember all unique destination ports for each pair of addresses separately. The source IP is a potential source of scan, meanwhile, the destination IP is a victim.

The algorithm was implemented as a module of the NEMEA system [1, 5] and is described in more detail in [4].

4 Evaluation and Measurements

Our measurement started on 31. 10. 2015 and stopped on 31. 12. 2015. In total, we observed over 388 billion flow records from all monitoring probes. That is on average 76,283 flows per second with over 144,506 flows per second in peak.

In order to find a reasonable threshold for the number of destination ports, we measured average number of destination ports used by a source IP. Moreover, we were interested in maximal number of destination ports per source. These observations were based only on TCP protocol without any consideration of TCP flags. Values were computed in hour intervals. The results are shown in Fig. 1. It is clear that intensive port scans probe a lot of destination ports. Therefore, the maximal number of ports is over fifty thousand. Most of source addresses use only a few destination ports and therefore total average number of destination ports per source IP lies around ten. From this point of view, with respect to memory consumption, the threshold was experimentally set to 50. Distribution function in Fig. 2 shows, that over 99 % addresses use less then 50 destination ports. Therefore, source IP address which has used 50 or more destination ports is considered as a potential attacker.

On average, network scans take about 1.2 % of observed flows. Alerts are aggregated in 10 minute time windows. Using the aggregation, the number of alerts decreases by 94 %. Average length of the aggregated alerts is about 2 minutes 45 seconds and over 2,600 destination ports are being probed. On average, there are over 580 aggregated alerts per hour.

During the analysis of scans targeted to a single target, we found distributed scans as well. Scanning hosts were active for about 10 minutes and each scanner probed about 2,000 ports. Altogether, scanners probed disjoint sets of ports.

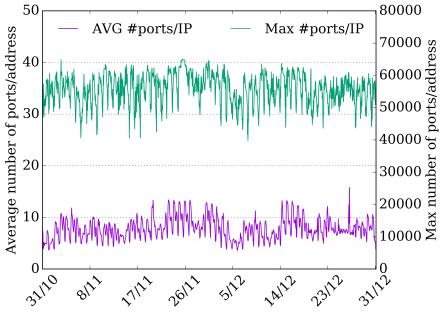

Fig. 1: Average and maximal number of destination ports per source IP.

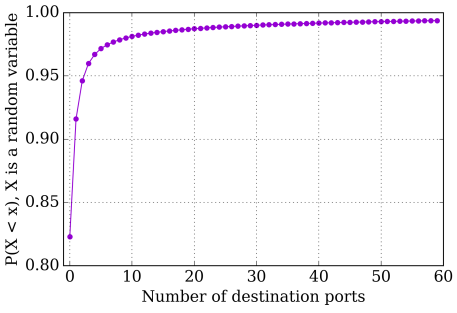

Fig. 2: Distribution function of number of destination ports per source IP.

According to the results, most of scans occur at 8:00 and 13:00 UTC. The rest of scans are spread over all hours. We expect that the distribution is caused by peaks of traffic that we normally see in these hours.

The deployed detector has discovered non-uniform intensity of some scans that was changing in time. For example, 47,156 addresses of Czech university of economics were scanned during 19 minutes by one block scan. The highest intensity (over 10,000 alerts per minute) was in the middle of the scan. Over 50 ports were probed for each target.

Memory consumption of the module was analyzed by valgrind [9]. The measurement was performed for almost two days and the module consumed 576 MiB during the peak. The amount of required memory is decreased due to autoregulation based on removing inactive addresses. It is set by module's threshold.

The used detection algorithm, from its nature, suffers from some limitation. The algorithm skips repeating SYN flows with the same destination port. Such traffic is assumed to be benign traffic. However, this fact can be easily exploited by scanners to avoid the detection. Distributed scans are generally difficult to detect. Large botnets can scan the whole internet using just a few packets generated by each bot [6]. A distributed scan can be detected by our algorithm if and only if at least some of the scanners fulfill conditions to be detected (e.g. number of destination ports threshold).

5 Conclusion

Vertical scans are frequent events that occur in almost every computer network. In this paper, we have proven this fact by observation of the vertical scans in the backbone network. The measurement showed that it is possible to detect scans with a simple straightforward detection algorithm using commodity hardware.

The proposed algorithm is limited to detection of TCP scans, however, it can be deployed in large network infrastructures and analyze huge volume of data. On average, there are about 580 aggregated alerts per hour that are detected by the implemented detection module. Some randomly selected alerts from the two-month measurement were verified manually. This paper presented statistical characteristics and results of scans detected at the perimeter of CESNET2.

Modern network attacks are mostly performed by botnets. Therefore, the importance of detection of distributed attacks (scans in our case) increases. According to our observations, intensive distributed scans became usual. However, the larger botnets are, the harder the detection is because each bot can probe just a few ports and so it is difficult to recognize bot's traffic from benign clients.

Acknowledgments This work was partially supported by the "CESNET E-Infrastructure" (LM2015042) and CTU grant No. SGS16/124/OHK3/1T/18 both funded by the Ministry of Education, Youth and Sports of the Czech Republic.

References

1. Bartoš, V., *et. al.*: Nemea: Framework for stream-wise analysis of network traffic. Tech. rep., CESNET, a.l.e. (2013), http://www.cesnet.cz/wp-content/uploads/2014/02/trapnemea.pdf
2. Bartos, V., Zadnik, M.: An analysis of correlations of intrusion alerts in an nren. In: Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2014 IEEE 19th International Workshop on. pp. 305–309. IEEE (2014)
3. Bhuyan, M.H., *et. al.*: Surveying port scans and their detection methodologies. The Computer Journal p. bxr035 (2011)
4. Cejka, T., Svepes, M.: Vertical Scan Detector README, https://github.com/CESNET/Nemea-Detectors/tree/master/vportscan_detector
5. CESNET, a. l. e.: NEMEA: Network Measurements Analysis Framework, https://github.com/CESNET/Nemea
6. Dainotti, A., *et. al.*: Analysis of a /0 stealth scan from a botnet. In: Proceedings of the 2012 ACM conference on Internet measurement conference. pp. 1–14. ACM (2012)
7. Jung, J., *et. al.*: Fast portscan detection using sequential hypothesis testing. In: Security and Privacy, 2004. Proceedings. 2004 IEEE Symposium on. pp. 211–225. IEEE (2004)
8. Lyon, G.F.: Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning. Insecure (2009)
9. Nethercote, N., Seward, J.: Valgrind: a framework for heavyweight dynamic binary instrumentation. In: ACM Sigplan notices. vol. 42, pp. 89–100. ACM (2007)
10. Paxson, V.: Bro: a system for detecting network intruders in real-time. Computer networks 31(23), 2435–2463 (1999)
11. Raftopoulos, E., *et. al.*: How dangerous is internet scanning? In: Traffic Monitoring and Analysis, Lecture Notes in Computer Science, vol. 9053, pp. 158–172. Springer International Publishing (2015)
12. Roesch, M., *et. al.*: Snort: Lightweight intrusion detection for networks. In: LISA. vol. 99, pp. 229–238 (1999)
13. Sridharan, A., Ye, T., Bhattacharyya, S.: Connectionless port scan detection on the backbone. In: Performance, Computing, and Communications Conference, 2006. IPCCC 2006. 25th IEEE International. pp. 10–pp. IEEE (2006)