International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 22 (2017) pp. 12652-12658

Storage Efficient Capturing of Port Scanning Attack Traffic

Rajni Ranjan Singh
Department of Computer Science and Engineering
Maulana Azad National Institute of Technology, Bhopal, M.P., India

Orcid Id: 0000-0002-8524-2556

Deepak Singh Tomar
Department of Computer Science and Engineering
Maulana Azad National Institute of Technology, Bhopal, M.P., India

Orcid Id: 0000-0001-9025-1679

Abstract-

Port is 16 bit number, it is associated with IP address to forms a socket which is useful to connect two applications remotely. Port scanning is a process to indentify open ports of remotely located host. Network administrator utilize port scanning to troubleshoot networking errors however attacker can perform port scanning to get information about target system like operating system details, open ports, firewall rules etc. Once open ports are identified, successively attacker may recognize vulnerability of associated applications, which can be further exploits. It is observed that approximately half of the cyber attacks preceded by port scanning attack. To identify source of the attack and to attribute attacker, forensic investigator analyzes the network packets in online/offline mode to identify type and source of the attack. However analyzing full network traffic dump is a time intense process therefore it is required to capture only relevant packets. In this work a storage efficient capturing method has been proposed to collect port scanning attack traffic. Here only relevant packets of port scanning attack are captured. Forensic investigator need not to analyze all the network traffic therefore analysis time will be reduced. To test and validate the effectiveness of the proposed method an experiment has been carried out using I & II Week DARPA Dataset of MIT Lincoln Laboratory, USA. It is observed that that if proposed method has been applied to real network environment than approximately 14 MB of network traffic data has been captured regularly(per day). The significant reduction in log size is about 96% of total traffic log size and 99.43% of total traffic (No. of packets).

Keywords— Port Scanning, Network Forensic, Intrusion Detection system.

INTRODUCTION

Network enumeration is a process to gather information about a network/host, this helps the attacker to find out the vulnerability of target system, so that it can be exploited. Port scanning is an enumeration technique. it can be categorized into vertical and horizontal based on the flow of packets from attacker to target systems. In horizontal scanning, same ports of all the network hosts are scanned. In vertical scanning all the port of a single host are scanned. Port scanning process may also be categorized based on the type of packers are utilized; details are given in the following sections.[1][2][4]

Network forensics is the process to capture, store and analyzes the network traffic to find out the source of the attack. Recording all the network traffic is not practically feasible because enormous amount of storage is required to store the network traffic. Preserving all the captured data on storage media is also a time consuming process. The analysis of captured data is the most critical and most time-consuming task as there is no full proof technique available for discriminating bogus traffic generated by an attacker from genuine traffic. Human judgment is also critical because with automated traffic examination tools, there is always a possibility of a false positive. Therefore it is paramount importance to research and adopt a method for capturing only relevant traffic rather than storing all the network traffic.[6][7][8]

RELATED WORK

Researchers have proposed various methods like Atul Kant Kaushik et al.[9] proposes a resource intensive forensic architecture consist of capturing, analysis and presentation modules. Proposed method captures the relevant packets from real network traces. However author mentioned that proposed work is limited to only TCP connect, FIN and ACK scan methods.

Rajni Ranjan Singh et al.[10] proposed a network forensic approach for stealth port scan attack. In this work they utilize signature based detection approach, An experiment has been carried out in real network environment using SNORT tool. However proposed work only focuses on the stealth type of port scanning methods.

Mehiar Dabbagh et al.[11] presented a methods for slow port scanning attack detection. Experiment has been carried out in

local area network environment against three most common scanning methods like TCP connect, Half open and FIN.

Omer Demir et al.[12] presented a session based packet logging and SYN based packet marking methods to reduce the storage requirement at the router. Various experiments have been carryout in real network environment. Authors mentioned that session based logging is useful for forensic investigation of network based crime cases.

PROPOSED WORK

Here an efficient packet capturing method has been proposed to capture only relevant port scan packets(traffic) instead of capturing all the network traffic. Eliminating bogus traffic helps the investigation to analyze the attack traffic efficiently. Port scanning attack can be categorized based on the types of packets are utilized. Various commonly used port scan techniques with associated features are given in the table I.

TABLE 1: PORT SCANNING TECHNIQUES CATEGORIZATION [3]

Port scanning Techniques Name	**Protocol used**	**TCP flag set**	**Victims relay, if port open**	**Victims replay, if port closed**
TCP Connect	TCP	SYN	ACK	RST
SYN scan		SYN	ACK	RST
SYN/ACK Scan		SYN/ACK	No	RST
FIN scan		FIN	No	RST
NULL Scan		No	No	RST
XMUS Scan		All Flag	No	RST

a) TCP Connect (Open scan): in this method, scanner establishes a connection with target computer using TCP/IP Three way handshaking mechanism. If destination port is opened, target responds by setting SYN and ACK flags of reply TCP packet.

Scanner -> SYN

Target -> SYN|ACK

Scanner -> ACK

If port is closed, target computer replied by Reset packet (A TCP Packet with RST flag set)

Scanner -> SYN

Target-> RST|ACK

Scanner-> RST

The entire connection attempt made by scanner machine has been logged at the destination application. Scanner can be easily caught by examine log entry of target application.

b) Syn Scan (Half open scanning): in this method scanner machine does not complete full handshaking mechanism therefore connection attempts are not logged at destination application. If destination port is opened than target computer responds with SYN /ACK packets.

Scanner -> SYN

Target -> SYN|ACK

Scanner -> RST

if port is closed Reset packet will be received

Scanner -> SYN

Target -> RST|ACK

c) SYN/ACK Scan: in this method scanner sends TCP packet with SYN/ACK flag set to the target computer. If Target port is opened than no response received otherwise Reset packet received.

Scanner -> SYN|ACK

Target -> No Response

If port is closed target responds with Reset packet.

Scanner -> SYN|ACK

Target -> RST

d) FIN Scan: This method is very much similar to SYN/ACK scan. If target port is opened, target computer do not send any packets.

Scanner -> FIN

Target -> -

If port is closed target responds with Reset packet.

Scanner -> FIN

Target -> RST

e) NULL Scan: in this scan type, scanner sends a TCP packet without setting any flags. Target does not reply, if target port is opened otherwise sends a Reset packet.

Scanner -> No Flag

Target -> -

If port is closed target responds with Reset packet.

Scanner -> No Flag

Target -> RST

f) XMUS Scan (Christmas Tree): this method is very much similar to NULL scan however in XMUS scan, all flags are set.

Scanner -> ALL Flag set

Target -> -

If port is closed target responds with Reset packet.

Scanner -> All Flag set

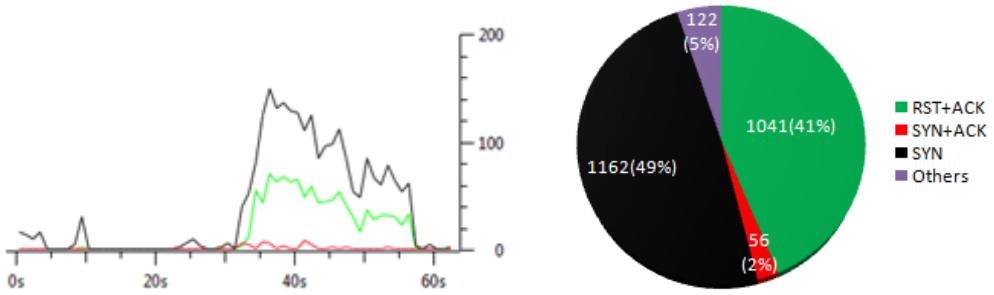
Target -> RST

Port scan methods like XMUS,NULL,SYN/ACK,FIN and SYN scan may not be detected by destination application because they does not establish full connection therefore these methods known as Stealth scanning. FIN, XMUS and NULL scanning methods are effective only on UNIX/LINUX operating systems.[1][3]

PORT SCAN ATTACK TRAFFIC ANALYSIS

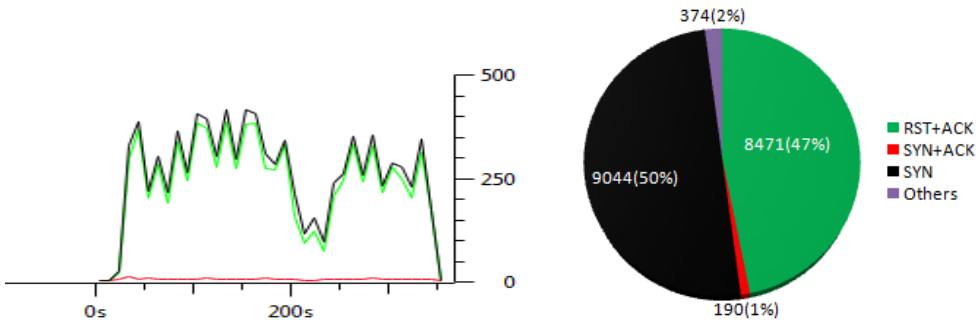
Port scan attack traffic analysis has been carried out by an experiment performed in real network environment. NMAP[13] is a well known port scanner tool has been utilized to perform attack and a packet sniffer has been configured to capture network traffic.

Following figures shows the distribution of various TCP packets and flags.



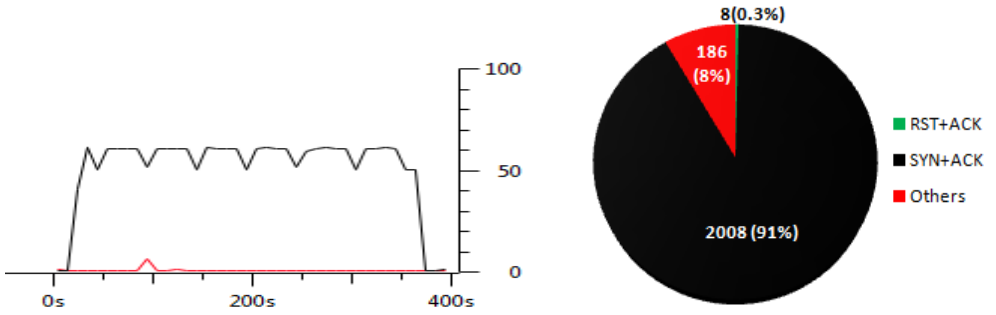
(a) Traffic flow (packet/second), X-axis: Time, Y-axis: Packets (b) Various TCP flags Distribution

Figure 1: SYN Scan(Half open scanning) attack traffic analysis



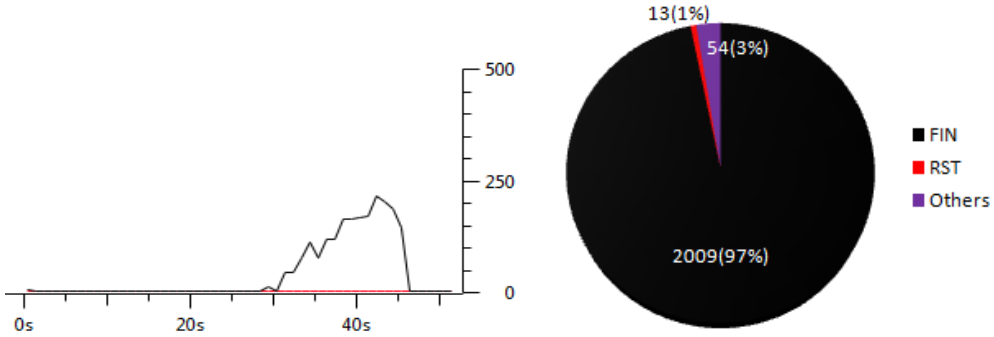
(a) Traffic flow (packet/second), X-axis: Time, Y-axis: Packets (b) Various TCP flags Distribution

Figure 2: TCP Connect scanning attack traffic analysis



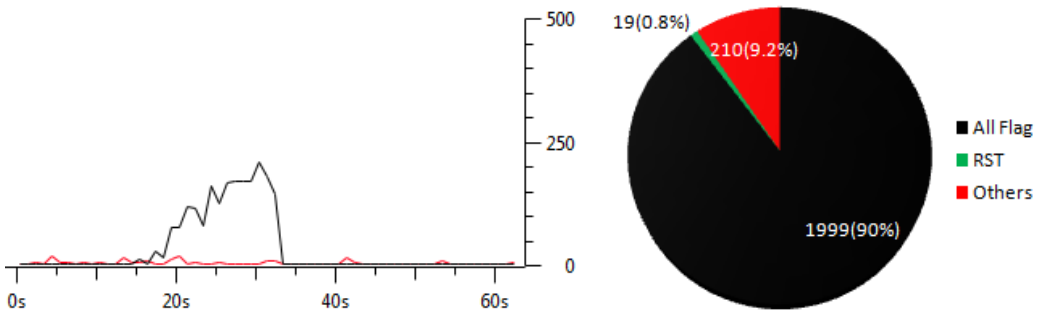
(a) Traffic flow (packet/second), X-axis: Time, Y-axis: Packets (b) Various TCP flags Distribution

Figure 3: SYN/ACK scanning attack traffic analysis



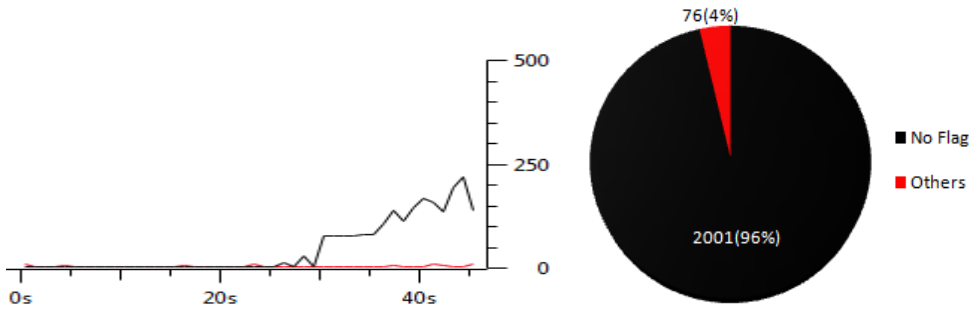
(a) Traffic flow (packet/second), X-axis: Time, Y-axis: Packets (b) Various TCP flags Distribution

Figure 4: FIN scanning attack traffic analysis



(a) Traffic flow (packet/second), X-axis: Time, Y-axis: Packet (b) Various TCP flags Distribution

Figure 5: XMUS scanning attack traffic analysis



(a) Traffic flow (packet/second), X-axis: Time, Y-axis: Packets (b) Various TCP flags Distribution

Figure 6: NULL scanning attack traffic analysis

As per the experiment carried out, observations are shown in table 2.

TABLE 2: OBSERVED NETWORK TRAFFIC

Port scanning Techniques Name	Total Packet Captured	Packet in Majority with flags (%)
TCP Connect	18079	SYN(50%), RST+ACK(47%)
SYN scan	2381	SYN(49%), RST+ACK(41%)
SYN/ACK Scan	2202	SYN+ACK(91%)
FIN scan	2076	FIN(97%)
NULL Scan	2228	ALL Flag(90%)
XMUS Scan	2077	No Flag(96%)

As per the information given in the table-2 common features are

PROTOCOL: TCP (TRANSMISSION CONTROL PROTOCOL)

Flags: SYN, RST+ACK, SYN+ACK, FIN, All and NULL.

Following algorithms is proposed to capture only relevant common feature and to eliminate bogus traffic. Finally Relevant_Packets_Vector contains all genuine port scanning traffic, which can be utilized for further analysis

Relevant Packet Capturing (RPC)

Variables used:

Packet i (flag): return the flag value of TCP packets

Packet i (Protocol): return the protocol of Network packet

Input: Incoming packets

Output: Relevent_Packet _Vector.

1. Initialize: Relevent_Packet_Vector [p_1,p_2……p_n] -> [0, 0….0];

2. Process incoming packet

3. if (Packet $_i$ (Protocol)= =TCP) then go to step 4 else go to step2.

4. if (Packet $_i$ (flag) == (SYN || (SYN && ACK) || (RST && ACK) || FIN || NULL || (RST && ACK && SYN && FIN && PUSH && URG)) then go to step 5 else go to step 2

5. Relevent_Packet _Vector. ->Packet i /* Add packet to vector*/ go to step 2.

Experimentation and Results

In order to analyze Effectiveness of RPC algorithm and their impact on storage requirement an experiment setup has been built in our academic network over a system having 1.4 GHz dual core processor with 4 GB RAM, A Snort IDSv2.9.4.6 [14] is installed and configured to run in NIDS(Network Intrusion detection system) mode with following rules using command "snort -i 1 -r c:\Dataset.dump -c c:\snort\etc\snort.conf -l c:\snort\log" here –r options of snort can be used to capture traffic from tcpdump file and –c used for NIDS mode. Following rules are formed to capture relevant packets.

```
alert tcp any any <> any any (msg:"NULL Scanning
set";flags:0;sid:7987660;)
alert tcp any any <> any any (msg:"SYN";flags:S;sid:7987661;)
alert tcp any any <> any any
(msg:"SYN+ACK";flags:SA;sid:7987662;)
alert tcp any any <> any any
(msg:"RST+ACK";flags:RA;sid:7987663;)
alert tcp any any <> any any (msg:"FIN";flags:F;sid:7987664;)
alert tcp any any <> any any
(msg:"ALL";flags:SARUPF;sid:7987665;)
```

Figure 7: SNORT Configured Rule set

Here I & II week network traffic data (tcpdump format) which is captured by The Cyber Systems and Technology Group (formerly the DARPA Intrusion Detection Evaluation Group) of MIT Lincoln Laboratory is processed as a testing data [15].

The above system implementation has been executed and the results are discussed here.

TABLE 3: CAPTURING SUMMARY EXPERIMENT DATA OF I & II WEEK DARPA DATASET

Experiment	All Packets	Logged Packets by	Size of all	Size of Logged
1	1,362,869	1,54,190	316243	11366
2	1,157,328	1,80,591	317769	13516
3	1,616,713	1,78,784	360134	13209
4	1,807,060	2,37,094	504924	17612
5	1,349,635	1,44,453	278101	10649
6	1,337,777	1,72,105	321604	12800
7	1,454,035	2,17,871	366991	16281
8	888,139	77,479	142284	5775
9	1,412,645	2,03,127	323110	15052
10	1,252,412	1,63,145	266891	12083
Total	13,638,61	77,479	3198051	128343

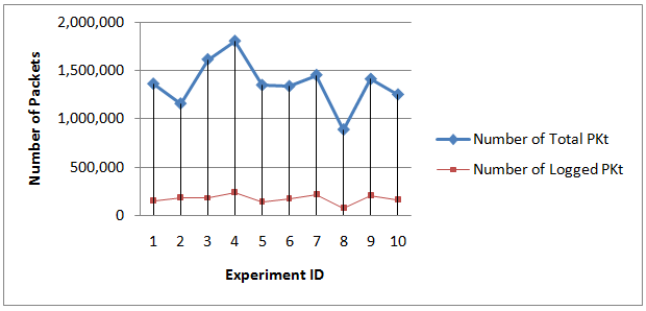
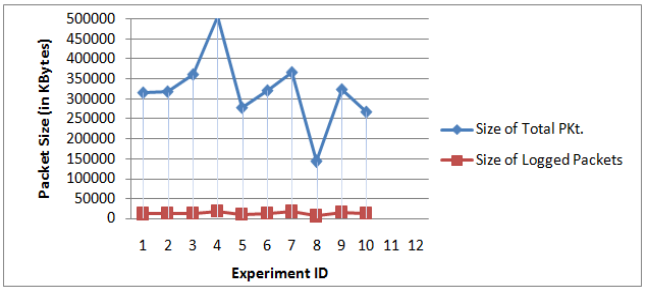


Figure 8: (a) Number of Packets Observed Over Different Experiments



8:(b) The Size of Packets (in KB) Observed Over Different Experiments

it is observed that due to Relevant packet Capturing (RPC) more than 99.43% bogus network traffic data is reduced that makes packet capturing faster and storage efficient. Using RPC the storage space consumed by the log files is reasonable (As shown in table-3 and figure-8). As per the experiments, it is observed that, for the network communication with a total data of 128343KB an experiment with about 217 hours in duration. Accordingly an average log size for the experiment should be about 128343/217 ×24 ≈ 14MB per day. This experiment method is suitable for regular use.

CONCLUSION

In this work a storage efficient packet capturing method has been introduced which is beneficial while investigating cyber attack preceded by any form of port scanning. Experimentation has been carried out using well known DARPA dataset and it is observed that if proposed method has been applied to real network environment than approximately 14 MB of network traffic data has been captured regularly(per day). Proposed method is beneficial to reduce storage requirement and analysis time of forensic investigation.

REFERENCES

[1] Elias Bou-Harb, Mourad Debbabi, and Chadi Assi, "Cyber Scanning: A Comprehensive Survey" IEEE Communications Surveys & Tutorials, Vol. 16, No. 3, Third Quarter 2014.

[2] Joanne Treurniet," A Network Activity Classification Schema and Its Application to Scan Detection" IEEE/ACM Transactions on Networking, Vol. 19, No. 5, October 2011

[3] whitepaper by dethy@synnergy.net "Examining port scan methods - Analyzing Audible Techniques"

[4] Susmit Panjwani, Stephanie Tan, Keith M. Jarrin, and Michel Cukier,"An Experimental Evaluation to Determine if Port Scans are Precursors to an Attack" Proceedings of the 2005 International Conference on Dependable Systems and Networks (DSN'05), IEEE-2005.

[5] Manowar H Bhuyan,D K Bhattacharyya and J K Kalita" Surveying port scans and their Detection methodologies" The Computer Journal (2011) 54 (10): 1565-1581 April 20, 2011.

[6] Emmanuel S. Pilli*, R.C. Joshi, Rajdeep Niyogi "Network forensic frameworks: Survey and research challenges" Digital Investigation Elsevier 7 (2 0 1 0) 1 4 – 2 4

[7] M.I. Cohen, PyFlag – An advanced network forensic framework , digital investigation 5 (2 0 0 8) S 1 1 2 – S 1 2 0 2008 Digital Forensic Research Workshop. Published by Elsevier Ltd.

[8] Bruce J. Nikkel, Improving evidences acquisition from live network sources, Digital investigation 3 (2006) 89–96, 2006 Elsevier Ltd.

[9] Atul Kant Kaushik, Emmanuel S. Pilli, R.C. Joshi, Network Forensic System for Port Scanning Attack, 2010 IEEE 2nd International Advance Computing Conference, 2010 IEEE

[10] Rajni Ranjan Singh and Deepak Singh Tomar, Network forensics: Detection and analysis of stealth port scanning attack, International Journal of Computer Networks and Communications Security, 3 (2), February 2015

[11] Mehiar Dabbagh, Ali J. Ghandour, Kassem Fawaz, Wassim El Hajj, Hazem Hajj, Slow Port Scanning Detection, 2011 7th International Conference on Information Assurance and Security (IAS), IEEE

[12] Omer Demir, Ping Ji and Jinwoo Kim, Session Based Logging (SBL) for IP-Traceback on Network Forensics Proceedings of the 2006 International Conference on Security & Management, SAM 2006, Las Vegas, Nevada, USA, June 26-29, 2006

[13] Namp https://nmap.org/

[14] Snort www.snort.org

[15] DARPA Dataset

http://www.ll.mit.edu/mission/communications/cyber/CSTcorpora/ideval/data/