International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 7 (2018) pp. 5291-5295

Effective Prevention Mechanism for IP flooding using Wireshark based on Network Traffic Analysis

Sathis Kumar[1*] Kavitha[2] Dinesh kumar[3] Mohan kumar[4]

[1, 2,3,4] *Assistant Professor, Department of CSE, Saranathan College of Engineering, Tamil Nadu, India.*
**Corresponding author*

Abstract

To prevent Denial of Service (DoS) attacks which prevents legitimate Cloud Users from accessing pool of resources provided by Cloud Providers by flooding and consuming network bandwidth to exhaust servers and all the resources. The main objective is to block the non-legitimate users, and prevent the network from flooding and protect the resources. The users who try to flood the network will compute to be stopped in the initial stage itself and they cannot access the resources completely. Non-legitimate users are identified based on threshold value, for the number of empty packets send through the network for a particular period of time. More over it provides the secured communication between the network and transport layer.

Keywords- TCP, Cloud User, Denial of Service, Firewall, IP Spoofing, IP, Threshold

INTRODUCTION

The DARPA model of the TCP/IP protocols consist of a four layers named as Application layer, Transport layer, Internet layer and Network Interface layer [1]. Each layer is equivalent to one or more layers of the OSI model.

An Overview of Wireshark

Wireshark is an open-source protocol or packet tracer. It runs on both Windows and UNIX platforms. Formerly known as Ethereal, its prime objective is network troubleshooting, analysis, and networking research. Wireshark facilitates a wide range of filters that supports over 1200 protocols (version 1.10.7); all with a simple front-end that enables one to break down the captured packets on the basis of different layers of the OSI (Open Systems Interconnection) model [2]. The Wireshark engine can decipher the structure of different networking protocols. This feature proves extremely beneficial for users to view the fields of each one of the headers and layers of the packets being analyzed [8]. Thus Wireshark provides a wide range of options to network engineers when performing certain traffic auditing tasks. Many tools, such as Snort, OSSIM and a number of IDS/IPS serve to warn users of some of the network related problems and attacks. In case of time deficiency, using these tools to analyze traffic in depth or monitor a network, lack in flexibility. Wireshark offers it very easily.

GRAPHICAL USAGE OF WIRESHARK

Wireshark like previously seen is a versatile tool that offers a wide range of options to examine the performance of a network graphically based on a multiple parameters. Graphical representation of statistics enables to study the network in a lucid manner [4].Two graphs in Wireshark prove to be extremely useful for traffic analysis. One of them is the Stevens graph and the other one is I/O graph. A TCP session can be tracked graphically to study the relationship between time and sequence number in a data stream. This graph is called a Time Sequence Graph (Steven) and can be found under the tab Statistics -> TCP Stream Graph. This graph can be extremely beneficial to detect irregularities in the behavior of TCP data flow. Another graph that gives valuable information about network traffic is the one on input/output. It can be found in Statistics - > I/O Graph [9]. Users can select various filters based on which they want to filter the data. It gives the graphical representation of various filters in different colours [10].

PROPOSED SYSTEM

The proposed system is capable of preventing attacks of various protocols like TCP, ARPDHCP, FTP, SNMP etc. It is based on a packet transfer threshold value. The threshold value is obtained by monitoring the packet transmission within the network for particular interval of time. The value is obtained by analyzing the server requests for a period of time. This system is mainly designed to identify the empty packet transmission and prevent the network from flooding.

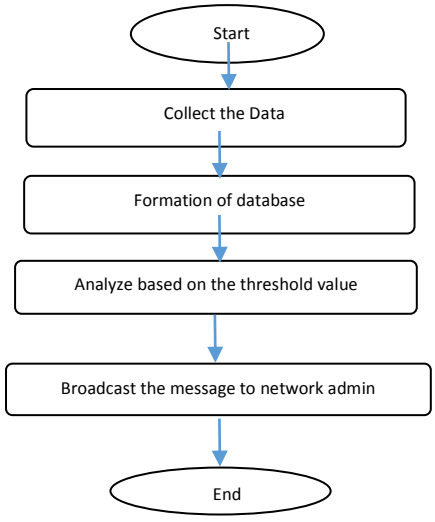

Figure 1. Flow diagram of the proposed system

Packet capturing is done using Wireshark and it is stored in .pcap format [12]. In the second module, captured information is converted into Xml file.Xml file can be easily converted into database [15].

IMPLEMENTATION

The very first step in auditing networks is to define where to analyze the traffic. Taking a common scenario for analysis, the following assumptions were made. There is a switched network made up of a number of switches, several terminals and a file server [5]. Network performance has dropped, however the cause is unknown. There are no IDS that can alarm or inform about attacks or network malfunction. Also, it is known that there are no problems with the transfer rate of the file server to LAN terminals [11].

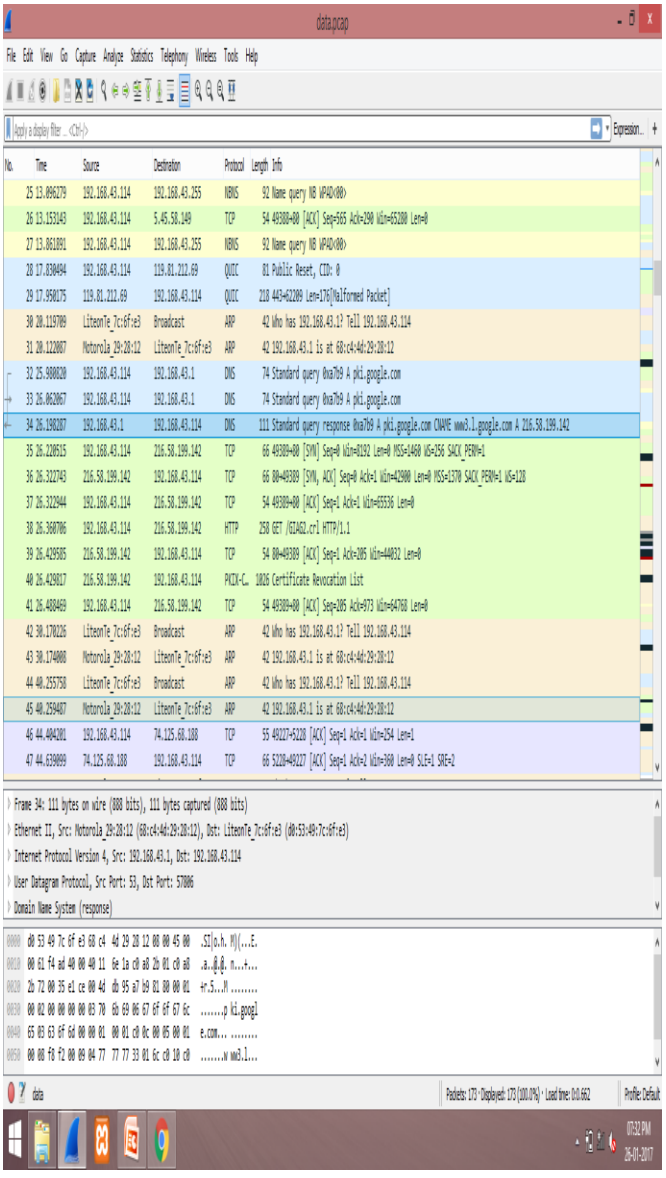

Figure 2 Capture Mode – hub connectivity between the server and the user's terminal where Wireshark is installed

How and Where to Capture the Data

The network equipment does not have Netflow protocols to analyze traffic remotely. Wireshark was chosen to analyze the above scenario. The first doubt which arises is where to install Wireshark. It would seem logical to install Wireshark on the file server itself to analyze the traffic that flows through this network segment. However, there could be situations in which there is no access to the server physically or quite simply for security reasons [16]. Thus, Wireshark cannot be installed there. Some alternatives are provided in the following paragraphs that enable to capture traffic without having to install Wireshark on the server [14].

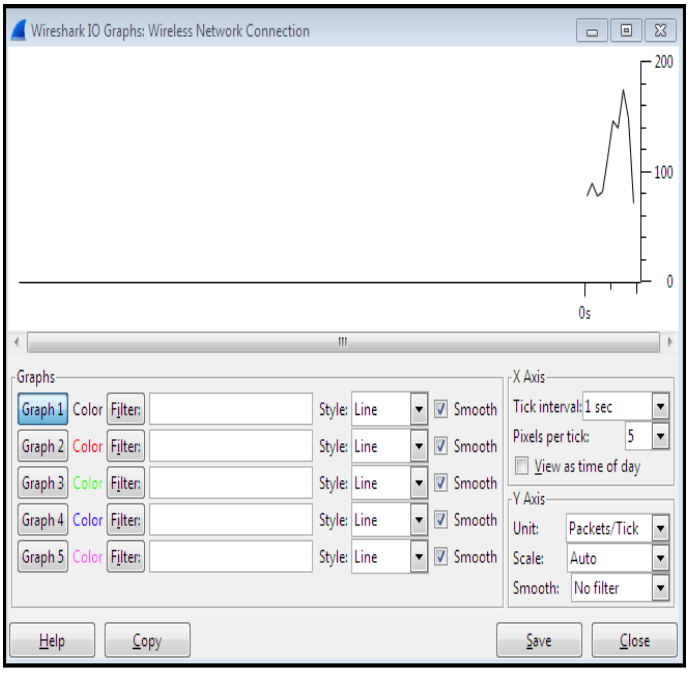

Figure 3 Wire shark IO (Input Output Graph) after data capturing

Converting Into Database

The captured data is in the raw format, to analyze the data we need to convert the .pcap file into xml format and then convert into database.

.pcap→xml→database

.pcap can be converted to XML format using wireshark. Then XML is converted into database using XAMPP.

Threshold

Threshold value is obtained by analyzing the packet transmission for a period of time within the network. By using the threshold value, filter the IP addresses that send number of packets more than the threshold value. Such IPs will be monitored and they will be blocked to access the network.

Wireshark->Firewall ACL Rules using wireshark firewall can be applied for any of the IP address to deny/allow packet from that particular IP.

The Figure 4 shows rules for the Firewall ACL (Access Control List) used to access/deny any product IPfilter, NetFilter, Packet Filter, IPFirewall, Windows Firewall for Filter.

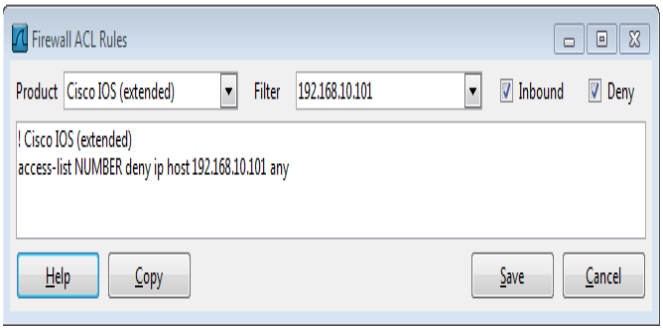

Figure 4. Firewall ACL rules

DDoS Attacks

In this attack, a large number of remotely controlled systems attack a single target, causing denial of service for users of the targeted system. The flood of incoming messages to the target system forces it to shut down, thereby denying service to the system to licit users. The attack that shut downs the site with the traffic itself, is called distributed denial-of-service (*DDoS*) attack [16]. As depicted, an Apache is installed on machine 10.0.0.101. A large number of TCP segments with the SYN flag activated from the same IP that do not receive a response (ACK) from the web service are generated [18].

In Wireshark, the packet sequence can be graphically seen by selecting from the menu *Statistics ->Flow Graph*.

This tool enables to track the behaviour of TCP connections as shown in Fig 5. It illustrates, using arrows, the source and target of each packet, highlighting the active flags that interfere in the connection flow. It can be easily noticed that there is a short period of time when a number of connection attempts are made by the IP 10.0.0.200 to port 80 of machine 10.0.0.101. This is a rather suspicious scenario [18]. The server has tried to resolve the MAC of the client many times (for example in packet 7852), but when no acknowledgement is received, it cannot send an ACK-SYN to the same machine to continue the three-step (handshake) connection. Thus the TCP/IP stack of the server has to wait for a set time for each connection [17]. During this idle time more packets keep arriving that trigger new connections. For each new connection, a structure in memory called the TCB (*Transmission Control Block*) is created and used by the TCP/IP stack of the operating system to identify each connection.

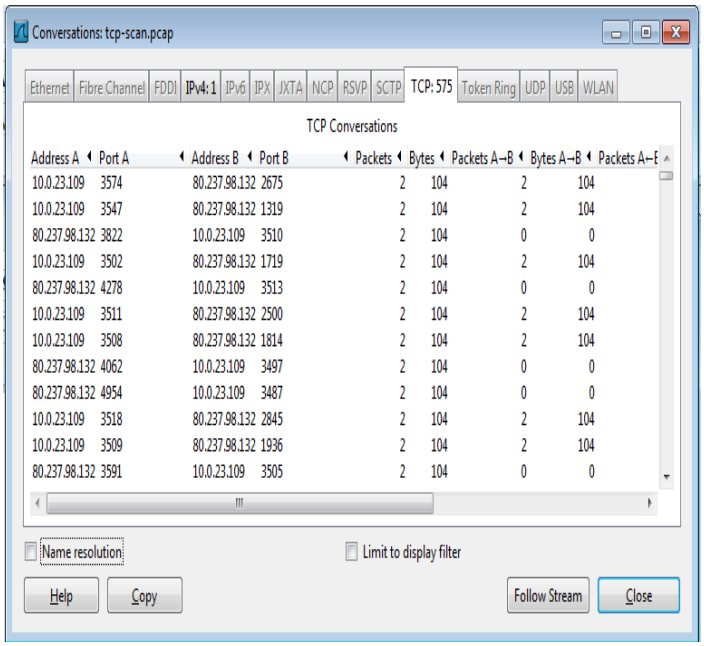

Figure 5 TCP Conversations with two end points

Wireshark->Flow Graph: It shows the communication between two or more different IP's.

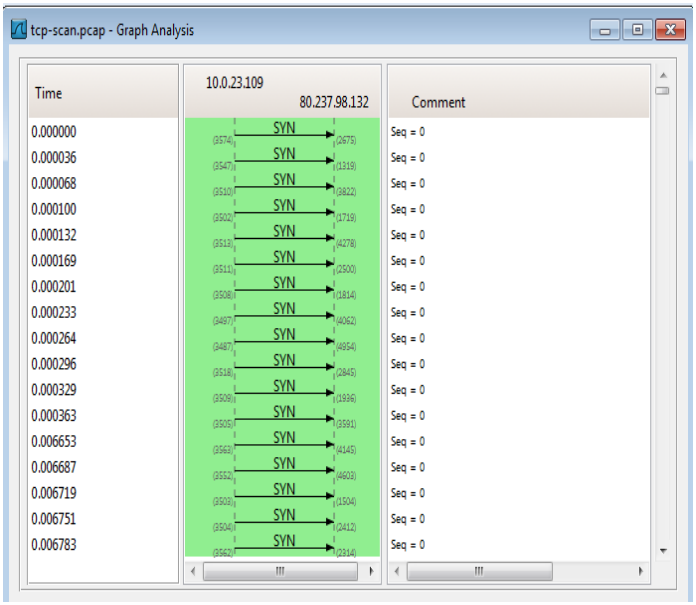

Figure 6 Flow graph

The message is transmitted from client to server using different different port number and no other message (SYN+ACK and ACK) is transmitted between the two IP .Therefore, connection is not established and finally DOS attack detected. In the above conversations as shown in Figure 6, we update this table by adding a column named as conclusion which may contain either the normal or abnormal value. Initially the value of that column is set to abnormal and then based on the condition of the normal behaviour conclusion column is updated. The condition explaining the

normal behaviour is the packet sent between port A and port B is a non-zero value indicating the connection establishment and data transfer[6]. Query used for this purpose is:

Update tablename set tablename.conclusion='normal' where tablename.PacketsA2B<>0AND tablename.PacketsB2A<>0

Where packetsA2B and packetsB2A are the packets sent from port A to port B and the packets sent from port B to port A respectively. After updating the table, we can apply any data mining technique such as association rule, classification, clustering etc to find the rules using either Weka or Rapid Miner Tool. Here, we have first applied classification using J48 algorithm which takes 0.09sec [19]. It is classified based on the number of packet sent, packet received and port number. Figure. 7 show the normal and abnormal behaviour of the ports with respect to the data transmission. The classification using J48 takes less time than random forest algorithm.

Broadcast

The IPs which are identified as anonymous, will be send to the admin and all the clients, who are connected to the network as a broadcast message.Admin and the clients will be aware of such users in future transmission.

- FF-FF-FF-FF-FF-FF Broadcast address: The packet having this address is received by all nodes and responded by them.
- FF-FF-FF-FF-FF-FE fake broadcast address: This address is fake broadcast address in which last 1 bit is missing. This address helps to verify whether the filter checks all the bits of address and respond to it.

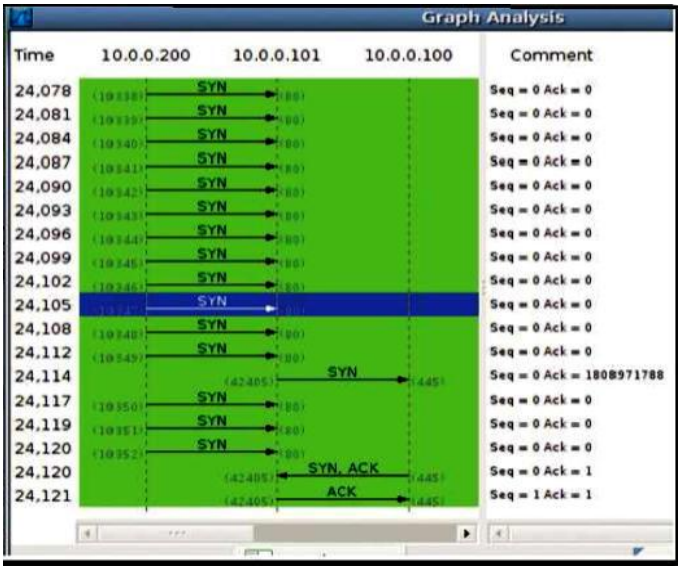

Figure 7. Data Flow Capture

- FF-FF-00-00-00-00 fake broadcast 16bit address: In this address we can see those first 16 bits are same as broadcast address.
- FF: 00:00:00:00:00 fake broadcast 8 bits: This address is fake broadcast address whose first 8 bits are same as the broadcast address.

CONCLUSION

Security Engineering deals with the prevention and detection of attacks launched over networks, particularly over the Internet. The problem is unlikely to be solved any time soon, as so many different kinds of vulnerability contribute to the attacker's toolkit. Ideally, people are expected to run written code on secure platforms carefully; in real life, this won't always happen. In this paper we try to concentrate on Transport and Network layer attack to make secure communication.

REFERENCES

[1] D.Adkins, K. Lakshminarayanan, A. Perrig, and I. Stoica. *Towards a more functional and secure network infrastructure.*Technical Report CSD-03-1242, UC Berkeley, 2003.

[2] D.G.Andersen. Mayday: Distributed filtering for Internet services. In USITS, Seattle, WA, 2003.

[3] T.Anderson, T.Roscoe, and D.Wetherall. Preventing Internet denial-of-service with capabilities.*In Proc. of Hotnets-II,* Cambridge, MA, Nov. 2003.

[4] T.Aura, P. Nikander, and J. Leiwo. DoS-resistant authentication with client puzzles. In SecurityProtocols—*8thInternational workshop, LectureNotesin Computer Science,* Cambridge, United Kingdom, Apr. 2000. Springer-Verlag.

[5] H.Balakrishnan, H. Rahul, and S. Seshan. An integrated congestion management architecture for Internet hosts. *In Proc.of ACM SIGCOMM '99*, Cambridge, MA, Sept. 1999.

[6] S.Bellovin, M. Leech, and T. Taylor. ICMP trace back messages, Internet draft, Work in progress, February 2003.

[7] CERT Advisory CA-97.28. IP denial-of-service attacks, Dec. 1997.

[8] D.Dean, M. Franklin, and A. Stubblefield. An algebraic approach to IP trace back. Information and System Security, 5(2):119–137, 2002.

[9] D.Dean and A. Stubblefield. Using client puzzles to protect TLS. *In Proc USENIX Security Symposium,* 2001.

[10] C.Dwork and M.Naor.Pricing via processing or combating junk mail. In E. Brickell, editor, Advances

in Cryptology — CRYPTO ’92, volume 740 of Lecture Notes in Computer Science, pages 139–147. *International Association for Cryptologic Research,* Springer-Verlag, 1993.

[11] V.D. Gligor. A note on the denial of service problem. *In Proc. of 1983 Symposium on Security and Privacy,* pages 139–149.IEEE, 1983.

[12] V.D. Gligor. Guaranteeing access in spite of service-flooding attacks. *In Proceedings of the Security Protocols Workshop*, Apr. 2003.

[13] H. Jamjoom and K. G. Shin. Persistent dropping: An efficient control of traffic aggregates. *In Proc. of ACM SIGCOMM ’03*,

[14] A.Juels and J.Brainard. Client puzzles: A cryptographic countermeasure against connection depletion attacks. *In Proc. of the Symposium on NDSS*, 1999.

[15] A. Keromytis, V. Misra, and D. Rubenstein. SOS: Secure Overlay Services. *In Proc. of ACM SIGCOMM,* Aug. 2002.

[16] R.Mahajan,S.M.Bellovin,S.Floyd,J.oannidis, V. Paxson, and S. Shenker. Controlling high bandwith aggregates in the network. *ACM Computer Communication Review,* 32(3):62– 73, July 2002.

[17] R.Merkle.Secure communication over insecure channels.Commun.ACM, 1(4):294–299, Apr. 1978

[18] S.Savage, D.Wetherall, A.Karlin, and T. Anderson. Practical network support for IP traceback.*In Proc. of ACM SIGCOMM 2000*, pages 295–306, Stockholm, and Sept. 2000.

[19] C. L.Schuba, I.V.Krsul, M. G. Kuhn, and E.H. Spafford.Analysis of a denial of service attack on TCP. *In Proc. Of IEEE Symposium on Security and Privacy,* May 1997

[20] C. Shields. What do we mean by network denial-of-service? *In Proc. of the 2002 IEEE Workshop on Information Assurance and Security*, West Point, NY, June 2002.