Tracing of Firewall and Denial of Firewall Attacks

[1] Prassanna Kumar M.J, [2]Vasantha H.T, [3]Vikas H.T,[4]Shwetha S, [5]Divya M.S

1Assistant Professor, Department of CSE, BGSIT, BG Nagar
2 Dept. of CSE, BGS Institute of Technology, BG Nagar, Karnataka, India
3Dept. of CSE, BGS Institute of Technology, BG Nagar, Karnataka, India
4 Dept. of CSE, BGS Institute of Technology, BG Nagar, Karnataka, India
5Dept. of CSE, BGS Institute of Technology, BG Nagar, Karnataka, India

Abstract-Firewalls are critical security devices handling all traffic in and out of a network. Firewalls, like other software and hardware network devices, have vulnerabilities, which can be exploited by motivated attackers. However, just like any other networking and computing devices, firewalls often have vulnerabilities that can be exploited by attackers. In this paper, first we investigate some possible firewall fingerprinting methods and surprisingly found that these methods can achieve quite high accuracy. Second, we study what we call Denial of Firewalling traffic to effectively overload a firewall. To our best knowledge, this paper represents the first study of firewall fingerprinting and Denial of Firewalling attacks.

I. INTRODUCTION

A. INSPIRATION

The security and reliability of firewalls are critical because they serve as the first line of defence in examining all traffic in and out of a network and they have been widely deployed for protecting both enterprise and backbone networks. However, just like any other networking and computing devices, firewalls often have vulnerabilities that can be exploited by attackers.To exploit firewall vulnerabilities, the first step that attackers need to do is firewall fingerprinting, i.e., identifying the particular implementation of a firewall including brand name, software/firmware version number, etc. For example, in the seminal work by Qian and Mao, the attacks discovered by them assumes that the attackers knows the particular implementation of the f,we study what we call Denial of Firewalling (DoF) attacks, where attackers use carefully crafted traffic to effectively overload a firewall. To our best knowledge, this paper represents the first study of firewall fingerprinting and Denial of Firewalling attacks. Designing counter measures for firewall finger printing and Denial of Firewalling attacks is out of the scope of this paper, but is the next step of this line of research.

B. LIMITATION OF PRIOR ART

There are several approaches to finding out the operating system ranging from simple banner observation to highly complicated TCP, UDP and ICMP header analysis. However, none of these methods can be used for firewall fingerprinting because firewalls, like other network middle boxes, forwards the traffic and cannot be targeted directly. For security purposes, some firewalls are configured in bridge mode with no IP address to be remotely accessible by the administrator. Hence, such approaches cannot be effective for firewall fingerprinting.

C. TECHNICAL CHALLENGE.

This work has three major technical challenges. First, finding the firewall implementation characteristics that we can use for fingerprinting is difficult because firewalls are mostly closed source and it is difficult to infer any implementation details from them. Moreover, there are many parameters and configuration details that can affect the performance of a firewall. Second, inferring the type of a target firewall is hard for attackers as they have no remote access to the firewall. Third, finding effective attack strategies on a firewall is difficult. Knowing some performance characteristics is not enough for designing effective attacks. Furthermore, for different firewall products, the DoF defence mechanisms may be different.

II. RELATED WORK

To the best of our knowledge, this is the first study of firewall fingerprinting and Denial of Firewalling attacks. The basic idea is to send packets that match the last rule in a firewall. However, it is extremely difficult, if not impossible, for an attacker to find the packets that match the last rule in a firewall without knowing the policy and implementation of the firewall. Work has also been done on firewall performance evaluation. Lyu and Lau measured the performance of a firewall under seven different policies. There are some industrial reports on

comparing commercial firewalls in terms of performance under different circumstances. Bosen in compared Secure Computing Side winder with Checkpoint's NGX and reported better throughput for Sidewinder when high-level of protection including packet and protocol inspection is required. Tolly Group, one of the independent test labs that performs extensive tests on different IT devices from different vendors, compared independent Checkpoint Firewall (VPN-1 Pro), PIX Firewall 535, and NetScreen-500. The report indicated that the Checkpoint Firewall outperform the other two firewalls in most of the tests run.

III. BACKGROUND

FIREWALL POLICIES

For each incoming or outgoing packet, a firewall decides to accept or discard it based on its policy. A firewall policy is composed of a sequence of rules, where each rule specifies a predicate over five different fields: source and destination port, source and destination IP address, and IP protocol.

PACKET CLASSIFICATION SOLUTIONS

The process of checking a packet against a firewall policy is called packet classification. Packet classification solutions fall into two main categories: software based solutions and Ternary Content Addressable Memory (TCAM) based solutions.

IV. OVERVIEW

MEASUREMENT ENVIRONMENT

Figure 1 shows the test bed topology for our testing of three different firewalls. Firewalls FW1 and FW2 are software firewalls running on a Linux machine with SMP kernel 2.6.Each firewall has 2 quad-core Intel Xeon 2.66GHz CPUs and16GB of RAM. FW3 is a hardware firewall that runs on a routing engine board with a 850MHz processor, 1,536MB DRAM, and 256MB compact flash. Each firewall is configured with the same policy comprised of 375 rules. The first 374 rules are set to accept traffic with the final rule discarding all traffic that is not specified previously. The firewall policy is chosen from real-life firewall policies used in a university campus network. The rules are defined over four packet header fields: source IP, destination IP, destination port number, and protocol. As with most real-life firewall policies, only a few rules overlap. Moreover, there is no rule hidden by another rule (i.e., there is no rule with lower index that completely covers a rule with higher index). Furthermore, the firewalls are only configured for packet filtering; other services such as VPN or NAT are disabled.

In addition to the firewalls, the testbed has two machines, VM1 and VM2, running VMW are ESX 3.5.0 on a similar machine with 2 quad-core Intel Xeon 2.66GHz CPUs and 16GB of RAM Each VMW are instance has four Linux virtual machines connected to each other by virtual switches. These virtual switches are connected directly (without an intermediary switch) to each firewall (FW1, FW2, and FW3). The virtual machines on VM1 and VM2 are used to place background traffic load on the firewalls by sending a substantial amount of packets to different interfaces of the firewall. The traffic is generated by Mausezahn network traffic generators (aka mz) [26], an open-source traffic generator. Using both VM1 and VM2, we are able to sustain a traffic rate of up to 300Mbps.Based on the design of experiments and attacks, the generated traffic can be accepted or discarded by the firewall to which it is sent. To put maximum load on the firewalls, the generated traffic has no packet payload. This maximizes the number of packets that a firewall needs to process. If packets have payloads, firewall throughput will increase, but traffic packet rate (i.e., packets per second) will decrease. As mentioned, the virtual switches are directly connected to the firewalls. This is to separate the generated traffic for each firewall and make firewall experiments independent from each other. The last portion of the testbed is the Probe Machine & Traffic Analyzer (PMTA): a Linux machine with Dual Quadcore Intel Xeon 2.66GHz CPUs and 16GB of RAM. We send probe packets by PMTA directly (i.e., no switch in between) to each firewall using an open-source packet generator hping2 [27]. If the probe packets are accepted by the target firewall they are routed back to PMTA through another interface (as it is shown in Figure 1). In order to measure firewall packet processing time, we use packet trace time-stamps. We use TCP dump [28] to dump packets with time- stamps with microsecond resolution. For the software firewalls (FW1 and FW2), we can analyze the packet traces and calculate the PPT based on the difference of packet trace time-stamps of outgoing and incoming interfaces. However, the hardware firewall (FW3) does not support TCP dump or any traffic monitoring (i.e., packet dumping) feature. Therefore, since we cannot measure the packet processing locally on the firewalls, the probe packets are forwarded to PMTA and we calculate the time-stamp difference of the packet traces on PMTA. The timestamp differences calculated on PMTA comprise the firewall PPT plus probe packet round trip

time (RTT) which in turn reduces the accuracy of firewall PPT.

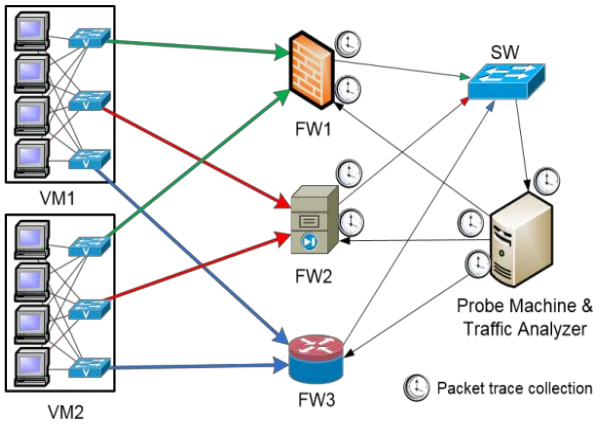


Fig.1. The test bed

V. TYPES OF FIREWALL

- Stateful
- Stateless

STATEFUL FIREWALL

Allow all traffic inbound with destination port 80Stateful firewallIn computing, a stateful firewall is a network firewall that tracks the operating state and characteristics of network connections traversing it. The firewall is configured to distinguish legitimate packets for different types of connections. Only packets matching a known active connection are allowed to pass the firewall. Stateful packet inspection (SPI), also referred to as dynamic packet filtering, is a security feature often included in business networks.

HISTORY

Stateful firewall technology was introduced by Check Point Software with the FireWall-1 product in 1994.Before the development of stateful firewalls, firewalls were stateless. A stateless firewall treats each network frame or packet individually. Such packet filters operate at the OSI Network Layer (layer 3) and function more efficiently because they only look at the header part of a packet.do not keep track of the packet context such as the nature of the traffic.Such a firewall has no way of knowing if any given packet is part of an existing connection, is trying to establish a new connection, or is just a rogue packet. Modern firewalls are connection-aware (or state-aware), offering network administrators finer-grained control of network traffic.The classic example of a network operation that may fail with a stateless firewall is the File Transfer Protocol (FTP). By design, such protocols need to be able to open connections to arbitrary high ports to function properly. Since a statelessfirewall has no way of knowing that the packet destined to the protected network (to some host's destination port 4970, for example) is part of a legitimate FTP session, it will drop the packet. Stateful firewalls with application inspection solve thisproblem by maintaining a table of open connections, inspecting the payload of some packets and intelligently associating new connection requests with existing legitimate connections.

DESCRIPTION

A stateful firewall keeps track of the state of network connections (such as TCP streams or UDP communication) and is able to hold significant attributes of each connection in memory. These attributes are collectively known as the state of the connection, and may include such details as the IP addresses and ports involved in the connection and the sequence numbers of the packets traversing the connection..Stateful inspection monitors incoming and outgoing packets over time, as well as the state of the connection, and stores the data in dynamic state tables. This cumulative data is evaluated, so that filtering decisions would not only be based on administrator-defined rules, but also on context that has been built by previous connections as well as previous packets belonging to the same connection.

The most CPU intensive checking is performed at the time of setup of the connection. Entries are created only for TCP connections or UDP streams that satisfy a defined security policy. After that, all packets (for that session) are processed rapidly because it is simple and fast to determine whether it belongs to an existing, pre-screened session. Packets associated with these sessions are permitted to pass through the firewall. Sessions that do not match any policy are denied, as packets that do not match an existing table entry In order to prevent the state table from filling up, sessions will time out if no traffic has passed for a certain period. These stale connections are removed from the state table. Many applications therefore send keep alive messages periodically in order to stop a firewall from dropping the connection during periods of no user-activity, though some firewalls can be instructed to send these messages for applications.

Depending on the connection protocol, maintaining a connection's state is more or less complex for the firewall. For example, TCP is inherently a stateful protocol as connections are established with a three- way handshake ("SYN, SYN-ACK, ACK") and ended with a

"FIN, FIN-ACK, ACK" exchange. This means that all packets with "SYN" in their header received by the firewall are interpreted to open new connections. If the service requested by the client is available on the server, it will respond with a "SYN-ACK" packet which the firewall will also track.

Once the firewall receives the client's "ACK" response, it transfers the connection to the "ESTABLISHED" state as the connection has been authenticated bi directionally. Application-level filters. Main article: Application firewall. Packet filtering alone is not regarded as providing enough protection. In order to effectively block peer-to-peer-related network traffic, what is needed is a firewall that does application filtering, which can be regarded as an extension to stateful packet inspection. Stateful packet inspection can determine what type of protocol is being sent over each port, but application-level filters look at what a protocol is being used for. For example, an application-level filter might be able to tell the difference between HTTP traffic used to access a Web page and HTTP traffic used for file sharing, whereas a firewall that is only performing packet filtering would treat all HTTP traffic equally.

Vulnerabilities

There is a risk that vulnerabilities in individual protocol decoders could allow an attacker to gain control over the firewall. This concern highlights the need to keep firewall software updated. Some stateful firewalls also raise the possibility that individual hosts can be tricked into soliciting outside connections. This possibility can only be completely eliminated by auditing the host software. Some firewalls can be defeated in this way by simply viewing a web page (either with JavaScript enabled, or after clicking on a button).Deny all traffic from 192.168.1.0/24 on the external interface.

Stateless protocol

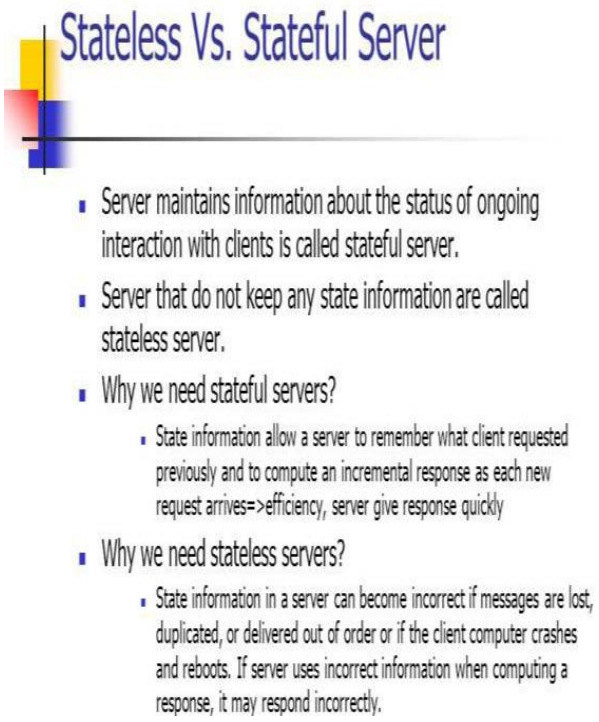
From Wikipedia, the free encyclopedia In computing, a stateless protocol is a communications protocol in which no information is retained by either sender or receiver. The sender transmits a packet to the receiver and does not expect an acknowledgment of receipt. A UDP connection-oriented session is a stateless connection because neither systems maintains information about the session during its life.

A stateless protocol does not require the server to retain session information or status about each communications partner for the duration of multiple requests. In contrast, a protocol that requires keeping of the internal state on the server is known as a stateful protocol. A TCP connection-oriented session is a 'stateful' connection because both systems maintain information about the session itself during its life.

Examples of stateless protocols include the Internet Protocol (IP), which is the foundation for the Internet, and the Hyper Text Transfer Protocol (HTTP), which is the foundation of data communication for the World Wide Web.

There can be complex interactions between stateful and stateless protocols among different protocol layers. For example, HTTP is an example of a stateless protocol layered on top of TCP, a stateful protocol, which is layered on top of IP, another stateless protocol, which is routed on a network that employs BGP, another statefulprotocol, to direct the IP packets riding on the network.

This stacking of layers continues even above HTTP. As a work-around for the lack of a session layer in HTTP, HTTP servers implement various session management methods, typically utilizing a unique identifier in a cookie or parameter that allows the server to track requests originating from the same client, and effectively creating a stateful protocol on top of HTTP.



VI. FIREWALL CHARACTERISTICS

To study firewall characteristics, we first give an overview on the methodology basics such as how the probe packets are sent and how the PPT is measured by

PMTA. We then show the results for different firewall features containing firewall packet classification algorithm, firewall statefulness and caching, and packets protocol and payload size impact

Methodology Basics

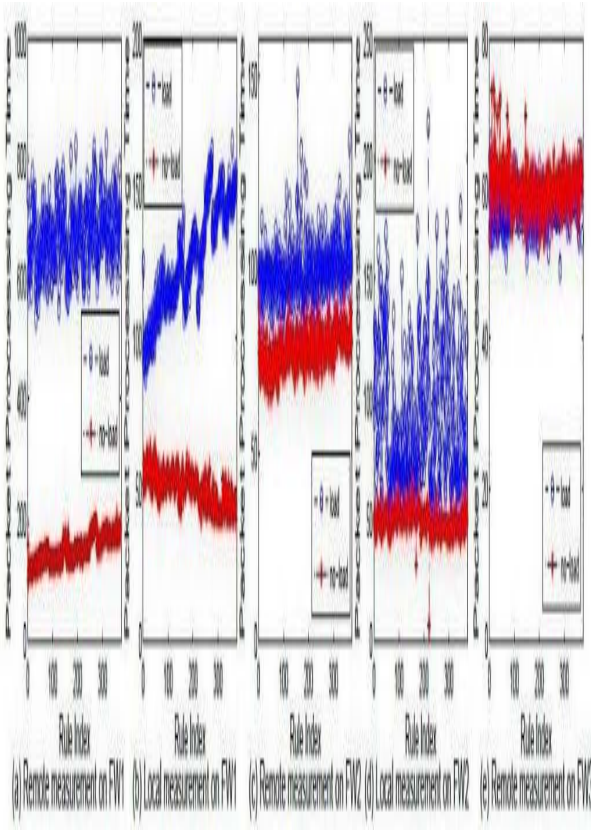
The probe packets are sent by the PMTA in four modes as follows:

• TCP Fix: A sequence of TCP packets with the same packet header.

• TCP Vary: A sequence of TCP packets with the same packet header except the source port which is chosen randomly for each probe packet.

• UDP Fix: A sequence of UDP packets with the same pacet header.

. UDP Vary: A sequence of UDP packets with the same packet header except the source port which is chosen randomly for each probe packet.



B .Impact of Packet Protocol and Payload Size

Firewalls usually perform queuing management techniques to improve their PPT. Such techniques can be made to be aware of the protocol and payload size of packets. In order to evaluate the impact of packet protocol and payload size, we configure all three firewalls in the stateless mode and repeat the same set of experiments while varying the packet payload size. Figure 4 shows the median PPT results for packet payload size of 0, 500, 1000, 1400 bytes.

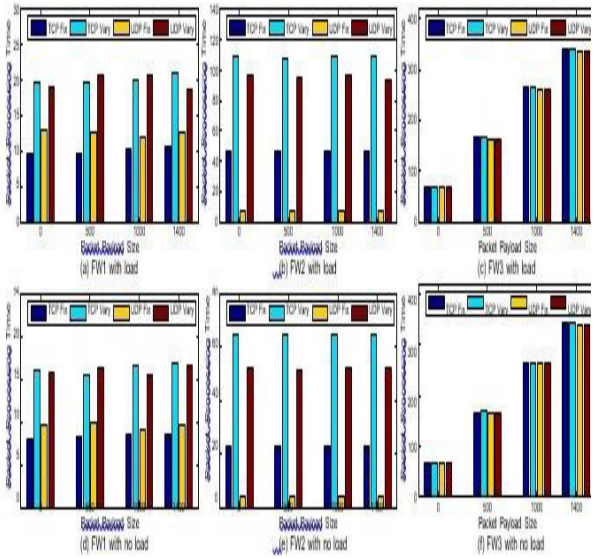


Fig2 The remote PPT for probe packets with different packet payload sizes

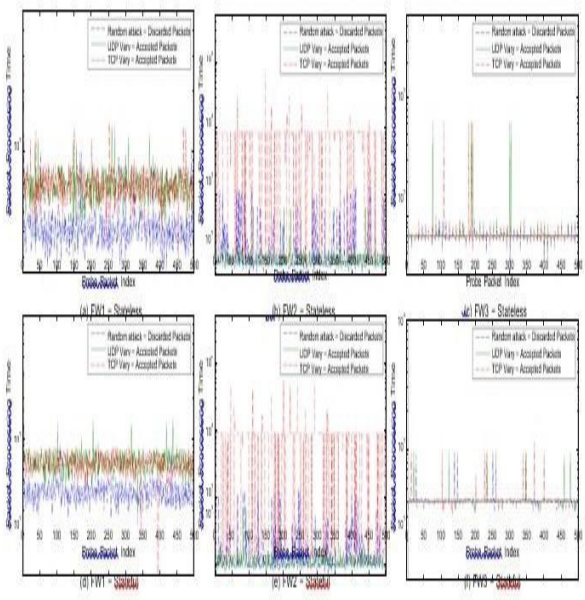
VII. ENIAL OF FIREWALLING ATTACKS

In order to effectively attack firewalls, we first use firewall characteristic measurements (conducted earlier in section V) to design effective customized attacks on the firewall. We then examine the effectiveness of the customized attacks by comparing the firewall performance under the customized attacks with the firewall performance under blind attacks. The experimental methodology herein is to create an attack scenario and monitor the firewall performance on legitimate traffic.

In our testbed setup, we drive attacks from all machines in VM1 and VM2 and send (legitimate) probe packets from the PMTA machine. We use the PPT observed by the probe as the performance indicator metric. We also use the CPU utilization on the firewall device as a measure of the firewall “stress” level. In our testbed, this CPU utilization information is available from FW1 and FW2, obtained through Simple Network Management Protocol (SNMP). FW3 does not provide access to this information.

In a blind attack, VM1 and VM2 send random UDP and TCP packets with no payload, which are mostly discarded by the firewall. In contrast, the customized attack packets are chosen to be accepted by the firewall.

The attack packets are generated in TCP Vary and UDP Vary modes. Figure 5 and Figure 6 show the packet processing time for 500 probe packets sent with one second interval when the firewalls are under blind attack, UDP Vary attack and TCP Vary attack. Note that in Figure 5 the customized attack packets have no payload, while in Figure 6 they have packet payload.



VIII. CONCLUSION AND FUTURE WORK

In this paper, I present methods for finding the firewall characteristics that are introduced by firewall implementations. Such characteristics can be exploited by attackers to identify black box firewalls with high accuracy and launch effective attacks on firewalls. We show two methods for inferring firewall implementation using these characteristics. The first method is based on the firewall decision on a sequence of TCP packets with unusual flags, which could be used as a firewall fingerprint for identification. The second method is based on machine learning techniques We further study the impact of different attacks on different firewalls and show that different firewalls are vulnerable to different attacks. While the best defense would involve working with firewall manufacturers to improve firewall implementations to minimize the impact of attacks, as future work we are willing to propose defense mechanisms from the firewall administrators' perspective, particularly in preventing attackers from gaining information about the firewall deployed and hence forcing attackers to use less-effective, blind attacks. Improved algorithm technique with a different simulation technique adopting the new technology and debugging the bugs.

REFERENCES

[1] Dan Goodin, "Hacker pierces hardware firewalls withwebpage",http://www.theregister.co.uk/2010/01/06/web based firewall attack/, 2010.
[2] Fyodor, "Nmap: Free network security scanner", http://nmap.org.
[3] Elizabeth Millard, "Best firewalls for big enterprises",http://www.ecommercetimes.com/story /business/21764.html, 2003.
[4]CheckPointFireWall-1,"http://www.checkpoint.com/" [19] Cisco˜PIX Firewalls,"http://www.cisco.com/",2011.[20]"Netscreen500",http://www.juniper.net/us/en/productsservices/end-of-sale/ns500/, 2008.