Survey of Development of Linux Firewall

[1]Rekha Pandey, [2]Aman Arora
[1]Research scholar, [2]Assistant Professor,
[1]Department of Computer Science Engineering
[1]Modern Institute of Engineering & Technology , Ambala, India

***Abstract* - In today's enterprise environment hackers and intruders have made many successful attempts to bring down high-profile company networks and web services. Many methods have been developed to secure the network infrastructure and communication over the Internet, like the use of firewalls, encryption, and virtual private networks. Using prevention systems, information can be collected from known types of attacks and find out if networks or hosts are being compromised. The information collected this way can be used to harden the network security, as well as it can be used to solve legal issues. Both commercial and open source products are available for this purpose.**

***Index Terms* - Firewall, Packet Filtering, Attack, IP**

I. INTRODUCTION

A computer network is just a set of stuff for nodes to talk to each other (`nodes' mean computers, printers, Coke machines etc.). It doesn't really matter how they are connected. Usually if you just connect two computers together, it's not called a network; you really need three or more to become a network. This is a bit like the word `group': two people is just a couple of guys, but three can be an `group'. Also, networks[2] are often hooked together, to make bigger networks; each little network (usually called a `sub-network') can be part of a larger network. The actual connection between two computers is often called a `network link'. If there's a bit of cable running out of the back of your machine to the other machines, that's your network link.

To keep unwanted intruders[6] off our computer network, we should install and configure a FIREWALL to separate the untrusted outside world from the trusted inside computer network. The firewall should inspect all network traffic and decide which traffic should be allowed to pass and which traffic should be blocked.

In order for all this to work, we have to tell the firewall what is acceptable network traffic by specifying policy rules. Every firewall has different methods of specifying what traffic is allowed to pass, and every firewall has different inspection possibilities. However, the basics of most firewalls are the same.

In this paper we explore the basics of firewalls, including a filtering strategy, packet filters, Network Address Translation (NAT), and application proxies.

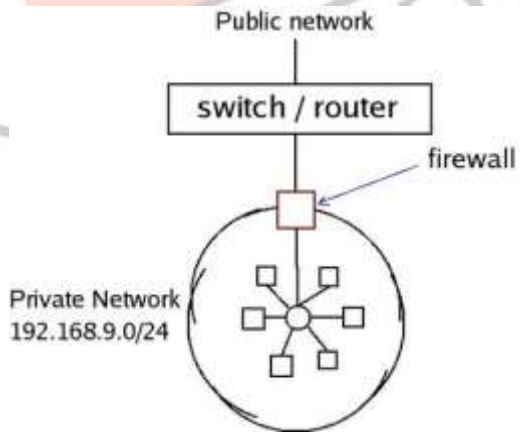

Fig. 1.1: Network/ Switch

II. FUNCTION OF FIREWALL

If ask several people what constitutes a firewall, you are bound to receive several different answers. Different firewall vendors use the term with different definitions. In its simplest form, a firewall is any device or software product sitting between your network and the Internet that blocks some network traffic. However, most people agree that a true firewall should have at least the following four basic functions:

2.1.1.1 Packet filtering: The headers of all network packets going through the firewall are inspected. The firewall makes an explicit decision to allow or block each packet.

2.1.1.2 Network Address Translation (NAT): The outside world sees only one or more outside IP addresses of the firewall. The internal network can use any address in the private IP address range. Source and destination addresses in network packets are automatically changed (or "translated") back and forth by the firewall.

2.1.1.3 Application Proxy: The firewall is capable of inspecting more than just the header of the network packets. This capability requires the firewall to understand the specific application protocol.

2.1.1.4 Monitoring and Logging: Even with a solid set of rules, logging what happens at the firewall is important. Doing so can help you to analyze a possible security breach later and gives feedback on the performance and actual filtering done by the firewall.

Because firewalls are a single point of entry for network traffic entering or leaving your internal network, the firewall is an excellent location to perform additional security tasks. Many firewalls support the following advanced functions:

2.1.1.5 Data caching : Because the same data or the contents of the same Web site may pass the firewall repeatedly in response to requests from different users, the firewall can cache that data and answer more quickly without getting the data anew from the actual Web site every time.

2.1.1.6 Content filtering: Firewall[4] rules may be used to restrict access to certain inappropriate Web sites based on URLs, keywords, or content type (video streams, for example, or executable e-mail attachments).

2.1.1.7 Intrusion detection: Certain patterns of network traffic may indicate an intrusion attempt in progress. Instead of just blocking the suspicious network packets, the firewall may take active steps to further limit the attempt, for example, by disallowing the sender IP address altogether or alerting an administrator.

2.1.1.8 Load balancing: From a security standpoint, a single point of entry is good. But from an availability standpoint, this single point of entry may lead to a single point of failure as well. Most firewalls allow the incoming and outgoing network request to be distributed among two or more cooperating firewalls.

Security threats that a firewall can't protect from are:

2.1.1.9 Inside attack: Users on the internal network have already passed the firewall. The firewall can do nothing to stop internal network snooping or intrusion attempts from within. Other security measures, such as configuring restricted permissions on workstations and servers, and enabling the auditing of network access, should be implemented to protect against these kinds of attacks. (Although you can deploy firewalls between your corporate servers and your internal users as well.)

2.1.1.10Social engineering: This is the term used to describe attacks in which hackers obtain information by calling employees and pretending to be a colleague at the front desk, a member of the security staff, or just somebody from the firm doing routine checks. This person asks for privileged information, such as server names, IP addresses, or passwords. Employees should be aware of these tactics and know that certain information should never be given.

2.1.1.11 Viruses and Trojan horse programs: Firewalls attempt to scan for viruses in all network traffic, but these wicked programs change constantly. Distinguishing between acceptable e-mail attachments and malicious content continues to be a problem for computer users. Good precautions should be taken to prevent the spread of viruses and to minimize the damage that a virus can do. Trojan horse programs are perhaps even harder to spot, because they don't attempt to spread to other files or computers like their virus sisters. A very small Trojan horse program that is run once by an unsuspecting user can open up a back door to his computer. A good example of the kind of damage that these programs can do is a Trojan horse program that sends out all collected keystrokes at password prompts once a week.

2.1.1.12 Poorly trained firewall administrators: The firewall doesn't know what is acceptable and what is not unless an administrator tells it. Competent firewall administrators should correctly specify which network traffic should be blocked. A Doorman has the intelligence to understand that a naked man who claims that his clothes and shoes already arrived and he is supposed to join them in the third floor conference room is clearly crazy, even though Doorman's security instructions may not have a naked-man-meeting-his-clothes-upstairs clause. Most firewalls, however, can easily be confused by fragmented IP packets and should be explicitly configured to handle such fragments.

New network protocols and services are introduced constantly. New vulnerabilities and software bugs in firewalls are also discovered constantly. Administrating a firewall is not a one-time task. Administrator should stay alert and constantly maintain the firewall rules, update and install vendor-supplied patches, and check the generated firewall log files.
Unfortunately, one can't just install a firewall and forget about it.

III. GENERAL STRATEGY

Allow-All or Deny-All

One of the first things that one must decide when one configure his firewall is the general strategy on how to specify what network packets and protocols you allow inside your network, and which network traffic that you want to block.

The two major possibilities are:

- Allow-all strategy: Allows all network packets except those that are explicitly denied.
- Deny-all strategy: Denies all network packets except those that are explicitly allowed.

At first sight, the Allow-all strategy[5] appears to be the easiest — requiring only that one create an exception list of network protocols or Web site content that is explicitly forbidden. This strategy is also in line with how other components work on network, such as non-firewall routers, network cards, and basically all computers that allow all traffic to pass except when explicitly denied.

The Allow-all strategy may sound enticing, but you should always use the second strategy — Deny-all, which is much more secure.

If Administrator uses the Allow-all strategy, he has to list every possible method that someone can use to intrude on his network and then come up with the rules to block related network traffic. Doing so results in a lot of rules, and even then Administrator is bound to miss one, two, or several methods that can be used to exploit your network. (Not to mention that list of deny rules would not include newly discovered methods or services that allow an intruder to enter your network.) This is akin to Doorman locking only the ground-floor windows that were previously used for illegal entrance into the building. Clearly, this is not a safe approach.

The Deny-all approach is much easier to administer. No traffic is allowed, except for a small number of explicitly defined protocols and services. The Deny-all approach has two advantages:

- Administrator has to maintain only a small list of allowed network traffic rules. The smaller the list, the easier it is for him to verify that the configuration of the firewall is correct.
- Administrator doesn't have to constantly add new rules to exclude newly discovered problems.

Note that your firewall may even use the Deny-all approach automatically. Of course, this means that if you haven't added any of the "Allow" rules yet, the firewall effectively blocks all network traffic. Somebody we knew participated in the test program of a new firewall product and was surprised at how many participants reported that the firewall seemed broken because they lost all network connectivity with the Internet after installing the product. They failed to realize that they had not yet added any "Allow" rules. (These were probably typical Allow-all people.)

Normally, the firewall policy that you want to express with firewall rules may actually be a combination of both Deny-all and Allow-all ingredients. The following firewall policy listing illustrates this point. In this example, the policy specifies what content network users can access on the Internet.

1. Deny network traffic on all IP ports.
2. Except, allow network traffic on port 80 (HTTP).
3. Except, from all HTTP traffic, deny HTTP video content.
4. Except, allow HTTP video content for members of the Trainers group.
5. Except, deny Trainers to download HTTP video content at night.

Packet Filtering

The first firewall products used only packet filtering to protect the internal network from outside users. The firewall inspected the IP header of each packet that entered the network and made a decision to allow or to block the packet based on the IP addresses used and the specific port number in the TCP or UDP header.

Although this functionality is still the cornerstone of firewall products, packet filtering used this way is not enough to secure the network. Packet inspection needs to be smarter about which inbound network packets[8] are expected in response to a legitimate request from an internal network user, and which inbound network packets are unsolicited and should therefore be blocked. When a firewall sees an outgoing network packet, it should remember that an incoming response is due soon, and only allow those expected incoming network packets. The remembered information is called state. This smarter form of packet filtering is called stateful packet filtering, as opposed to the original stateless packet filtering, which did not remember the state of expected return packets.

Even with stateful packet filtering, networks still have a couple of vulnerabilities that should be addressed:

- The outside world can learn the IP addresses used on the internal network. The firewall should use Network Address Translation (NAT) to solve this problem.

- Packet filters have limited decision capabilities because they look only at a small portion of the network packet. The firewall should use application proxy functionality to further inspect the packet.

IV. FILTERING IP DATA

Packet filters are rules that inspect the information in the packet header of every network packet arriving at the firewall, so that they can decide whether the packet should be allowed in or out or whether it should be dropped.

If the packet is allowed to pass, it continues on its merry way. But note that an IP packet never passes any router or firewall without undergoing some modifications. Before the IP packet is sent on its way, the router or firewall reduces the numerical value of the Time-To-Live (TTL) information in the IP header by at least 1. If the TTL value, which the sender of the packet probably set at 128, ever reaches 0, the packet is discarded. Discarding the packet prevents endless looping of IP packets in cyberspace, due to router misconfiguration or similar causes. Because the TTL field in the IP header changes, the value of the Header checksum field must be recalculated and is changed as well.

The IP packet may be modified even more when passing through the firewall. Later in this chapter, we show you how to add Network Address Translation (NAT) to the firewall. In NAT, the IP numbers and port numbers that are used in the packet are substituted with other numbers before the packet continues. When the firewall also performs application proxy functionality, the network packet may not pass at all, and instead, be rebuilt from scratch and sent along.

Administrator can create packet filter rules that check the following fields in a network packet that arrives at the firewall:

4.1Source IP address: This is the IP address that the packet lists as its sender. This field doesn't necessarily reflect the true original computer that sent the packet. The field may have been changed for legitimate reasons by a NAT machine between the sender and the firewall, or hackers may have changed the field, which is known as IP spoofing.

4.2 Destination IP address: This is the IP address to which the packet is being sent. Make sure you list the actual IP address in the packet filter rule and not the Domain Name System (DNS) name, such as server3. dummies.com. Otherwise, a hacker that takes over a DNS server can immediately pass all packet filters undisturbed.

4.3 IP protocol ID: An IP header can be followed by different protocol headers. Each of these protocols has its own IP protocol ID. The best-known examples are TCP (ID 6) and UDP (ID 17). Others that you will encounter are ICMP (ID 1), GRE (ID 47) — which is used for PPTP connections — and ESP (ID 50) and AH (ID 51), which are both used for the IPSec protocol.

4.4TCP or UDP port number: The port number indicates to which service this packet is destined. You should allow only ports that are associated with allowed services, such as HTTP (port 80) or FTP (port 20/21). The Appendix contains a list of many well-known port numbers and their associated services.

4.5 ICMP message type: ICMP is the housekeeping protocol of the TCP/IP protocol suite. Some of the ICMP types are very useful messages; others are very dangerous and should not be allowed to pass through the firewall.

4.6 Fragmentation flags: IP packets can be broken into smaller packets to accommodate network segments that can only handle smaller-sized packets. Unfortunately, as is discussed a later in the chapter, this functionality can be misused.

4.7 IP Options setting: Optional functions of TCP/IP can be specified in this field. Hackers can exploit the Source Route option in particular. These options are only used for diagnostics, so the firewall should drop network packets with IP Options set.

Besides checking the fields mentioned in the preceding list, packet filters can also make a distinction between packets that are outbound from the internal network to the Internet, and inbound from the Internet to the internal network. The network interface on which a packet arrives is an important criterion by itself. Because a hacker can easily forge the sender IP address in a network packet, the firewall can't really trust that information. However, if the packet arrives at the external network interface using a sender IP address that belongs to the internal network, the firewall should recognize the IP spoof immediately, just by noticing that it arrives at the external network interface. In the next few sections, we will give further details on how packet filters can inspect the information in the packet headers.

4.8ICMP

Many ICMP packets are useful in diagnosing network connectivity. The bestknown example is the PING application that sends an ICMP Echo Request to another machine. If that machine is available, an ICMP Echo Reply packet is returned to the PING application. Other useful ICMP types are TTL Exceeded and Destination Unreachable, which indicate that the packet did not reach the final destination.

Table 4.1 shows sample ICMP packet filters that allow PING from the internal network.

ICMP Packet Filters

Protocol	*Type*	*Direction*	*Action*
ICMP	Echo Request	Outbound	Allow
ICMP	Echo Reply	Inbound	Allow
ICMP	TTL Exceeded	Inbound	Allow
ICMP	Destination Unreachable	Inbound	Allow
ICMP	Echo Request	Inbound	Deny
ICMP	Echo Reply	Outbound	Deny
ICMP	Redirect	Inbound	Deny

The direction in the packet filter is important because it distinguishes between the PING command that is initiated on the internal network (Allow) and the externally initiated PING command (Deny).

V. CONCLUSION

During the entire of our project duration the focus of our development was of the functionality and we kept it towards how our solution can actually come into application rather than just restricting it into particular domain of interest so this is why that this application is not developed taking into account issues like how much it will be user-friendly.

REFERENCES

] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, *Computer Networks*, 38 (2002), 393–422.
[2] N. A.Pantazis, S. A. Nikolidakis, D. D. Vergados, Energy-efficient routing protocols in wireless sensor networks: A Survey, *IEEE Communications Surveys and Tutorials*, Vol. 15, No. 2, Second Quarter, 2013, 551-591.
[3] W. Guo, W. Zhang, A survey on intelligent routing protocols in wireless sensor networks, *Journal of Network and Computer Applications*, 38 (2014), 185–201.
[4] S. M. Zin, N. B. Anuar, L. M. Kiah, A. K. Pathan, Routing protocol design for secure WSN: Review and open research issues, *Journal of Network and Computer Applications*, 41 (2014) 517–530.
[5]A. WilligandH. Karl, Protocols and architectures for wireless sensor networks. John Wiley and Sons Ltd.(2005).
[6]I.F. Akyildiz, W. Su, Y.Sankarasubramaniam, E. Cayirci, A survey on sensor networks, *Communications magazine, IEEE*, 40 (8), 102-114.
[7] I.F. Akyildiz, M.C. Vuran, Wireless sensor networks, John Wiley & Sons Ltd, 2010.
[8] F. Hu, X. Cao, Wireless sensor networks: Principles and Practice, CRC Press, 2010