Firewall (computing)

From Wikipedia, the free encyclopedia

In computing, a **firewall** is a network security system that monitors and controls the incoming and outgoing network traffic based on predetermined security rules.[1] A firewall typically establishes a barrier between a trusted, secure internal network and another outside network, such as the Internet, that is assumed not to be secure or trusted.[2] Firewalls are often categorized as either *network firewalls* or *host-based firewalls*. Network firewalls filter traffic between two or more networks; they are either software appliances running on general purpose hardware, or hardware-based firewall computer appliances. Host-based firewalls provide a layer of software on one host that controls network traffic in and out of that single machine.[3][4] Firewall appliances may also offer other functionality to the internal network they protect, such as acting as a DHCP[5][6] or VPN[7][8][9][10] server for that network.[11][12]

Contents

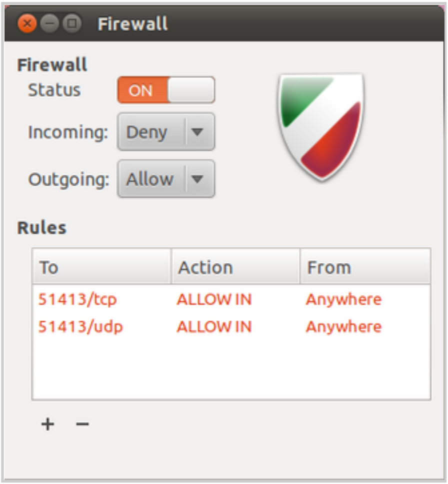
History

The term *firewall* originally referred to a wall intended to confine a fire or potential fire within a building.[13] Later uses refer to similar structures, such as the metal sheet separating the engine compartment of a vehicle or aircraft from the passenger compartment.

The term was applied in the late 1980s to network technology that emerged when the Internet was fairly new in terms of its global use and connectivity.[14] The predecessors to firewalls for network security were the routers used in the late 1980s.[15]

First generation: packet filters

The first type of network firewall was the packet filter which would look at network addresses and ports of the packet to determine if that packet should be allowed or blocked.[16] The first paper published on firewall technology was in 1988, when engineers from Digital Equipment Corporation (DEC) developed filter systems known as packet filter firewalls. This fairly basic system was the first generation of what is now a highly involved and technical internet security feature. At AT&T Bell Labs, Bill Cheswick and Steve Bellovin were continuing their research in packet filtering and developed a working model for their own company based on their original first generation architecture.[17]



Firewall

Firewall
Status: ON
Incoming: Deny
Outgoing: Allow

Rules

To	Action	From
51413/tcp	ALLOW IN	Anywhere
51413/udp	ALLOW IN	Anywhere

Screenshot of Gufw: The firewall shows its settings for incoming and outgoing traffic.

Packet filters act by inspecting the "packets" which are transferred between computers on the Internet. If a packet does not match the packet filter's set of filtering rules, the packet filter will drop (silently discard) the packet or reject it (discard it, and send "error responses" to the source). Conversely, if the packet matches one or more of the programmed filters, the packet is allowed to pass. This type of packet filtering pays no attention to whether a packet is part of an existing stream of traffic (i.e. it stores no information on connection "state"). Instead, it filters each packet based only on information contained in the packet itself (most commonly using a combination of the packet's source and destination address, its protocol, and, for TCP and UDP traffic, the port number). TCP and UDP protocols constitute most communication over the Internet, and because TCP and UDP traffic by convention uses well known ports for particular types of traffic, a "stateless" packet filter can distinguish between, and thus control, those types of traffic (such as web browsing, remote printing, email transmission, file transfer), unless the machines on each side of the packet filter are both using the same non-standard ports.[18]

Packet filtering firewalls work mainly on the first three layers of the OSI reference model, which means most of the work is done between the network and physical layers, with a little bit of peeking into the transport layer to figure out source and destination port numbers.[19] When a packet originates from the sender and filters through a firewall, the device checks for matches to any of the packet filtering rules that are configured in the firewall and drops or rejects the packet accordingly. When the packet passes through the firewall, it filters the packet on a protocol/port number basis (GSS). For example, if a rule in the firewall exists to block telnet access, then the firewall will block the TCP protocol for port number 23.[20]

Second generation: "stateful" filters

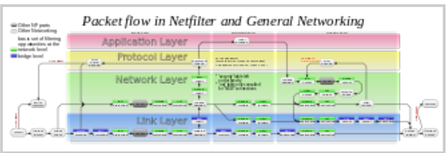
From 1989–1990 three colleagues from AT&T Bell Laboratories, Dave Presotto, Janardan Sharma, and Kshitij Nigam, developed the second generation of firewalls, calling them circuit-level gateways.[21]

Second-generation firewalls perform the work of their first-generation predecessors but operate up to layer 4 (transport layer) of the OSI model. This is achieved by retaining packets until enough information is available to make a judgement about its state.[22] Known as stateful packet inspection, it records all connections passing through it and determines whether a packet is the start of a new connection, a part of an existing connection, or not part of any connection.[23] Though static rules are still used, these rules can now contain *connection state* as one of their test criteria.

Certain denial-of-service attacks bombard the firewall with thousands of fake connection packets in an attempt to overwhelm it by filling its connection state memory.[24]

Third generation: application layer

Marcus Ranum, Wei Xu, and Peter Churchyard developed an application firewall known as Firewall Toolkit (FWTK). In June 1994, Wei Xu extended the FWTK with the kernel enhancement of IP filter and socket transparent. This was known as the first transparent application firewall, released as a commercial product of Gauntlet firewall at Trusted Information Systems. Gauntlet firewall was rated one of the top firewalls during 1995–1998.



Flow of network packets through Netfilter, a Linux kernel module

The key benefit of application layer filtering is that it can "understand" certain applications and protocols (such as File Transfer Protocol (FTP), Domain Name System (DNS), or Hypertext Transfer Protocol (HTTP)). This is useful as it is able to detect if an unwanted application or service is attempting to bypass the firewall using a protocol on an allowed port, or detect if a protocol is being abused in any harmful way.

As of 2012, the so-called next-generation firewall (NGFW) is nothing more than the "wider" or "deeper" inspection at application stack. For example, the existing deep packet inspection functionality of modern firewalls can be extended to include

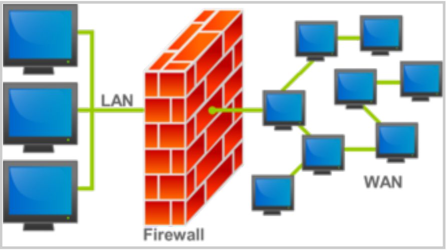
- Intrusion prevention systems (IPS)
- User identity management integration (by binding user IDs to IP or MAC addresses for "reputation")
- Web application firewall (WAF). WAF attacks may be implemented in the tool "WAF Fingerprinting utilizing timing side channels" (WAFFle)[25]

Types

Firewalls are generally categorized as network-based or host-based. Network-based firewalls are positioned on the gateway computers of LANs, WANs and intranets. Host-based firewalls are positioned on the network node itself. The host-based firewall may be a daemon or service as a part of the operating system or an agent application such as endpoint security or protection. Each has advantages and disadvantages. However, each has a role in layered security.

Firewalls also vary in type depending on where communication originates, where it is intercepted, and the state of communication being traced.[26]

Network layer or packet filters



An illustration of where a firewall would be located in a network

Network layer firewalls, also called packet filters, operate at a relatively low level of the TCP/IP protocol stack, not allowing packets to pass through the firewall unless they match the established rule set. The firewall administrator may define the rules; or default rules may apply. The term "packet filter" originated in the context of BSD operating systems.

Network layer firewalls generally fall into two sub-categories, stateful and stateless. Stateful firewalls maintain context about active sessions, and use that "state information" to speed packet processing. Any existing network connection can be described by several properties, including source and destination IP address, UDP or TCP ports, and the current stage of the connection's lifetime (including session initiation, handshaking, data transfer, or completion connection). If a packet does not match an existing connection, it will be evaluated according to the ruleset for new connections. If a packet matches an existing connection based on comparison with the firewall's state table, it will be allowed to pass without further processing.

Stateless firewalls require less memory, and can be faster for simple filters that require less time to filter than to look up a session. They may also be necessary for filtering stateless network protocols that have no concept of a session. However, they cannot make more complex decisions based on what stage communications between hosts have reached.

Newer firewalls can filter traffic based on many packet attributes like source IP address, source port, destination IP address or port, destination service like HTTP or FTP. They can filter based on protocols, TTL values, network block of the originator, of the source, and many other attributes.

Commonly used packet filters on various versions of Unix are *ipfw* (FreeBSD, Mac OS X (< 10.7)), *NPF* (NetBSD), *PF* (Mac OS X (> 10.4), OpenBSD, and some other BSDs), *iptables*/*ipchains* (Linux) and *IPFilter*.

Application-layer

Application-layer firewalls work on the application level of the TCP/IP stack (i.e., all browser traffic, or all telnet or FTP traffic), and may intercept all packets traveling to or from an application. They block other packets (usually dropping them without acknowledgment to the sender).

On inspecting all packets for improper content, firewalls can restrict or prevent outright the spread of networked computer worms and Trojans. The additional inspection criteria can add extra latency to the forwarding of packets to their destination.

Application firewalls function by determining whether a process should accept any given connection. Application firewalls accomplish their function by hooking into socket calls to filter the connections between the application layer and the lower layers of the OSI model. Application firewalls that hook into socket calls are also referred to as socket filters. Application firewalls work much like a packet filter but application filters apply filtering rules (allow/block) on a per process basis instead of filtering connections on a per port basis. Generally, prompts are used to define rules for processes that have not yet received a connection. It is rare to find application firewalls not combined or used in conjunction with a packet filter.[27]

Also, application firewalls further filter connections by examining the process ID of data packets against a rule set for the local process involved in the data transmission. The extent of the filtering that occurs is defined by the provided rule set. Given the variety of software that exists, application firewalls only have more complex rule sets for the standard services, such as sharing services. These per-process rule sets have limited efficacy in filtering every possible association that may occur with other processes. Also, these per-process rule sets cannot defend against modification of the process via exploitation, such as memory corruption exploits. Because of these limitations, application firewalls are beginning to be supplanted by a new generation of application firewalls that rely on mandatory access control (MAC), also referred to as sandboxing, to protect vulnerable services.[28]

Proxies

A proxy server (running either on dedicated hardware or as software on a general-purpose machine) may act as a firewall by responding to input packets (connection requests, for example) in the manner of an application, while blocking other packets. A proxy server is a gateway from one network to another for a specific network application, in the sense that it functions as a proxy on behalf of the network user.[2]

Proxies make tampering with an internal system from the external network more difficult, so that misuse of one internal system would not necessarily cause a security breach exploitable from outside the firewall (as long as the application proxy remains intact and properly configured). Conversely, intruders may hijack a publicly reachable system and use it as a proxy for their own purposes; the proxy then masquerades as that system to other internal machines. While use of internal address spaces enhances security, crackers may still employ methods such as IP spoofing to attempt to pass packets to a target network.

Network address translation

Firewalls often have network address translation (NAT) functionality, and the hosts protected behind a firewall commonly have addresses in the "private address range", as defined in RFC 1918. Firewalls often have such functionality to hide the true address of protected hosts. Originally, the NAT function was developed to address the limited number of IPv4 routable addresses that could be used or assigned to companies or individuals as well as reduce both the amount and therefore cost of obtaining enough public addresses for every computer in an organization. Although NAT on its own is not considered a security feature, hiding the addresses of protected devices has become an often used defense against network reconnaissance.[29]

See also

- Access control list
- Air gap (networking)
- Bastion host
- Comparison of firewalls
- Computer security
- De-perimeterisation
- Distributed firewall
- Egress filtering
- End-to-end principle
- Firewall pinhole
- Firewalls and Internet Security
- Golden Shield Project
- Guard (information security)
- Identity-based security
- IP fragmentation attacks
- List of Unix-like router or firewall distributions
- Mangled packet
- Mobile security § Security software
- Next-Generation Firewall
- Personal firewall
- Screened-subnet firewall
- Unidirectional network
- Unified threat management
- Virtual firewall
- Vulnerability scanner
- Windows Firewall

References

1. Boudriga, Noureddine (2010). *Security of mobile communications*. Boca Raton: CRC Press. pp. 32–33. ISBN 0849379423.
2. Oppliger, Rolf (May 1997). "Internet Security: FIREWALLS and BEYOND". *Communications of the ACM*. **40** (5): 94. doi:10.1145/253769.253802 (https://doi.org/10.1145%2F253769.253802).
3. Vacca, John R. (2009). *Computer and information security handbook*. Amsterdam: Elsevier. p. 355. ISBN 9780080921945.
4. "What is Firewall?" (https://personalfirewall.comodo.com/what-is-firewall.html). Retrieved 2015-02-12.
5. "Firewall as a DHCP Server and Client" (https://paloaltonetworks.com/documentation/70/pan-os/pan-os/networking/firewall-as-a-dhcp-server-and-client.html). *Palo Alto Networks*. Retrieved 2016-02-08.
6. "DHCP" (http://www.shorewall.net/dhcp.htm). *www.shorewall.net*. Retrieved 2016-02-08.
7. "What is a VPN Firewall? - Definition from Techopedia" (https://www.techopedia.com/definition/30753/vpn-firewall). *Techopedia.com*. Retrieved 2016-02-08.
8. "VPNs and Firewalls" (https://technet.microsoft.com/en-us/library/cc958037.aspx). *technet.microsoft.com*. Retrieved 2016-02-08.
9. "VPN and Firewalls (Windows Server)" (https://technet.microsoft.com/en-us/library/cc753364%28v=ws.10%29.aspx). *Resources and Tools for IT Professionals | TechNet*.
10. "Configuring VPN connections with firewalls" (http://www.techrepublic.com/article/configuring-vpn-connections-with-firewalls/).

11. Andrés, Steven; Kenyon, Brian; Cohen, Jody Marc; Johnson, Nate; Dolly, Justin (2004). Birkholz, Erik Pack, ed. *Security Sage's Guide to Hardening the Network Infrastructure*. Rockland, MA: Syngress. pp. 94–95. ISBN 9780080480831.
12. Naveen, Sharanya. "Firewall" (https://www.paloaltonetworks.com/documentation/glossary/what-is-a-firewall). Retrieved 7 June 2016.
13. Canavan, John E. (2001). *Fundamentals of Network Security* (1st ed.). Boston, MA: Artech House. p. 212. ISBN 9781580531764.
14. Liska, Allan (Dec 10, 2014). *Building an Intelligence-Led Security Program*. Syngress. p. 3. ISBN 0128023708.
15. Ingham, Kenneth; Forrest, Stephanie (2002). "A History and Survey of Network Firewalls" (http://www.cs.unm.edu/~treport/tr/02-12/firewall.pdf) (PDF). Retrieved 2011-11-25.
16. Peltier, Justin; Peltier, Thomas R. (2007). *Complete Guide to CISM Certification*. Hoboken: CRC Press. p. 210. ISBN 9781420013252.
17. Ingham, Kenneth; Forrest, Stephanie (2002). "A History and Survey of Network Firewalls" (http://www.cs.unm.edu/~treport/tr/02-12/firewall.pdf) (PDF). p. 4. Retrieved 2011-11-25.
18. TCP vs. UDP By Erik Rodriguez (http://www.skullbox.net/tcpudp.php)
19. William R. Cheswick, Steven M. Bellovin, Aviel D. Rubin (2003). "Google Books Link (https://books.google.com/books?id=_ZqIh0IbcrgC&lpg=PA142&dq=Firewalls%20and%20Internet%20Security%2C%20by%20Cheswick%20et%20al.&pg=PA176#v=onepage&q=Firewalls%20and%20Internet%20Security,%20by%20Cheswick%20et%20al.&f=false)". *Firewalls and Internet Security: repelling the wily hacker*
20. Aug 29, 2003 Virus may elude computer defenses (https://news.google.com/newspapers?id=neIqAAAAIBAJ&sjid=Vo4EAAAAIBAJ&pg=4057,6607496&dq=firewall&hl=en) by Charles Duhigg, Washington Post
21. *Proceedings of National Conference on Recent Developments in Computing and Its Applications, August 12–13, 2009* (https://books.google.com/books?id=TnJk09xmdFsC&pg=PA513&lpg=PA513&dq=circuit+level+gateways+at%26t&source=bl&ots=AJ1qvKxvGF&sig=4RcxAO2-bENP2fbzIeSreghVe9E&hl=en&sa=X&ei=g-5WU6qGGMmyyAS-lYC4DA&ved=0CDYQ6AEwAg#v=onepage&q=circuit%20level%20gateways%20at%26t&f=true). I.K. International Pvt. Ltd. 2009-01-01. Retrieved 2014-04-22.
22. Conway, Richard (204). *Code Hacking: A Developer's Guide to Network Security*. Hingham, Massachusetts: Charles River Media. p. 281. ISBN 1-58450-314-9.
23. Andress, Jason (May 20, 2014). *The Basics of Information Security: Understanding the Fundamentals of InfoSec in Theory and Practice* (2nd ed.). Elsevier Science. ISBN 9780128008126.
24. Chang, Rocky (October 2002). "Defending Against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial". *IEEE Communications Magazine*. **40** (10): 42–43. doi:10.1109/mcom.2002.1039856 (https://doi.org/10.1109%2Fmcom.2002.1039856).
25. "WAFFle: Fingerprinting Filter Rules of Web Application Firewalls" (https://www.usenix.org/conference/woot12/waffle-fingerprinting-filter-rules-web-application-firewalls). 2012.
26. "Firewalls" (http://www.tech-faq.com/firewall.html). MemeBridge. Retrieved 13 June 2014.

27. "Software Firewalls: Made of Straw? Part 1 of 2" (http://www.symantec.com/connect/articles/software-firewalls-made-straw-part-1-2). Symantec Connect Community. 2010-06-29. Retrieved 2014-03-28.
28. "Auto Sandboxing" (http://help.comodo.com/topic-72-1-451-4846-.html). Comodo Inc. Retrieved 2014-08-28.
29. "Advanced Security: Firewall" (http://technet.microsoft.com/en-us/library/hh831365.aspx). Microsoft. Retrieved 2014-08-28.

External links

- Internet Firewalls: Frequently Asked Questions (http://www.faqs.org/faqs/firewalls-faq/), compiled by Matt Curtin, Marcus Ranum and Paul Robertson.
- Firewalls Aren’t Just About Security (http://www.cyberoam.com/downloads/Whitepaper/ApplicationFirewall.pdf) - Cyberoam Whitepaper focusing on Cloud Applications Forcing Firewalls to Enable Productivity.
- Evolution of the Firewall Industry (http://docstore.mik.ua/univercd/cc/td/doc/product/iaabu/centri4/user/scf4ch3.htm) - Discusses different architectures and their differences, how packets are processed, and provides a timeline of the evolution.
- A History and Survey of Network Firewalls (http://www.cs.unm.edu/~treport/tr/02-12/firewall.pdf) - provides an overview of firewalls at the various ISO levels, with references to the original papers where first firewall work was reported.
- Software Firewalls: Made of Straw? Part 1 (http://www.securityfocus.com/infocus/1839) and Software Firewalls: Made of Straw? Part 2 (http://www.securityfocus.com/infocus/1840) - a technical view on software firewall design and potential weaknesses

Retrieved from "https://en.wikipedia.org/w/index.php?title=Firewall_(computing)&oldid=793412535"

- This page was last edited on 1 August 2017, at 15:49.