Firewall systems

Joaquin Garcia-Alfaro

Recommended minimum reading time: 3 hours

Contents

Introduction

Firewall systems represent an effective element in preventing cyber attacks. Already consolidated as indispensable elements to guarantee the protection of computers and computer networks, firewall systems represent a practical implementation of the concept of *access control*, both at the system level (for example, to prevent attacks against personal computers at the scale of application) and at the network level (to prevent attacks from hostile networks against personal or corporate networks).

This module assumes that you have a basic knowledge of how computer networks operate and some notions of cyber security. More precisely, and although there is no need for extensive knowledge, we consider that you are already familiar with the Open Systems Interconnection (OSI) reference model, as well as with the TCP/IP family of protocols.

In this module, we present a general introduction to firewall systems. One of the main concepts related to these systems is *packet filtering*. The first firewall systems (currently known as *first generation firewalls*) are actually routers using filtering rules to build an entry barrier and thus give way to what is known as *perimeter security*. The aim is to separate vulnerable environments, generally private networks, from hostile environments (for example, public networks, such as the Internet). Thus, in this module, we will first deal with firewall systems based on packet filtering, responsible for processing and inspecting traffic at the network layer level (level 3 of the OSI reference model). Then, we will also see configuration examples of the second and third generation firewall systems, responsible for filtering traffic with a more thorough inspection (levels 4 and 7 of the OSI reference model). Finally, we will discuss other aspects related to firewall systems that may be of interest to you, such as the most used current architectures, as well as the advantages and limitations of implementing perimeter security and its deployment in different types of networks using firewall systems.

Objectives

The aims to be achieved with this material are the following:

1. To know what a firewall system is, and to understand how it can be used to provide protection to a computer network, thus preventing computer attacks.

2. To know and understand the limitations of firewall systems.

3. To know and understand some strategies and architectures related to firewall systems, starting with information filtering.

4. To understand what perimeter security policies are and their deployment on the network using firewall system architectures.

1. Introduction to firewall systems

Firewall systems are hardware or software components that control traffic in and out of a system. In general, they are usually used to provide an access control mechanism on computer networks and they allow separating an internal part (in which the computers involved are considered as trusted) from other computers located in the outside (potentially hostile).

A **firewall system** is responsible for separating computer networks and controlling the traffic that circulates there. The control consists of allowing, denying or redirecting communications from one of the networks to the other, through an access control or use of the associated network protocols.

Local firewalls

Although in this module we refer to firewall systems as elements to control traffic on a computer network, the same concept can be applied to attack prevention by installing a local firewall to directly protect computers and devices.

Therefore, a firewall system serves as a barrier in a network. It can be used to block incoming or outgoing traffic, to prevent unauthorized access, etc. In this context, the concept of a security policy is important. In other words, a firewall system makes it possible to implement the security policy associated with the network.

A firewall system is one of the possible security mechanisms that allow implementing the rules of a security policy. More specifically, the rules relating to access control at the network level and which are related to the perimeter security of the network.

When installing and configuring a firewall system, the following must be kept in mind:

- All traffic leaving or entering the network must pass through the firewall system. This can be achieved by physically blocking all access within the network.

- Only authorized traffic, defined in the system's local security policies, will be able to bypass the block.

- The firewall itself must be protected against possible attacks or intrusions.

Firewall systems, as we know them today, appeared at the end of the eighties, developed by the DEC and AT&T companies. In 1991, the first commercial firewall, the DEC SEAL, appeared. Today, firewall systems are a very important element not only in network devices, but even in personal computers. There are different technologies for implementing firewalls, and above all, there are many architectures or ways to configure firewalls in a network. In this module we will see some of the most prominent ones. It is important to note that we will focus on the use of firewalls in TCP/IP networks, although the use of firewalls is not exclusive to these particular protocols.

Complementary reading

The following article (available online) discusses the origins of firewall systems in more detail: **Frédéric Avolio** (June 1999). "Firewalls and Internet Security", The Internet Protocol Journal (volume 2, number 2, pages 24-32). <http://ipj.dreamhosters.com/wp-content/uploads/issues/1999/ipj02-2.pdf>

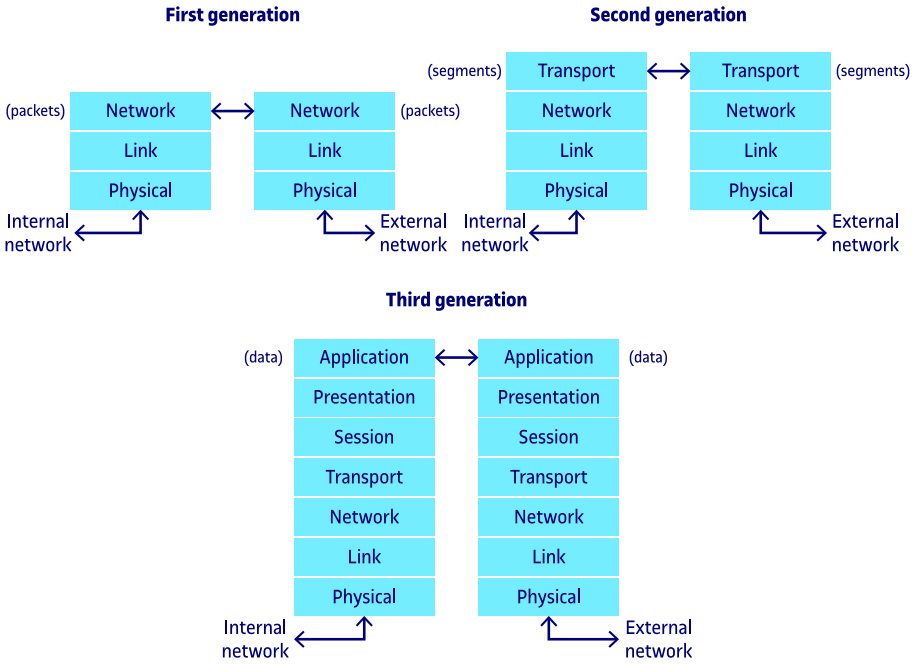
Likewise, it is important to always keep in mind what we want to achieve with a firewall system. The most common use is to control incoming and outgoing traffic from one network to another. Generally, it is about protecting the internal network of an organization against a hostile network, such as the Internet. In the internal network we can find personal computer equipment, printers, mobile devices, smartphones, servers, etc. The presence of servers, if they offer services to the external network, may require special treatment in firewall systems. We refer, for example, to web, email or file sharing servers that the organization wants to offer on the Internet. In general, these servers are not treated like personal computers when designing their protection.

As we will see in the following sections, although firewalls provide many security measures, it should be noted that they are not a definitive or unique solution to the problem of network security. There are many threats that cannot be covered by firewall systems. In this sense, a very important aspect, as we will see, is that it is difficult to protect against an internal attacker with a firewall. The firewall itself, like any computer system, can present zero-day vulnerabilities and be a victim of malicious programs within the operating system in which it runs. In addition, firewall systems can have a certain degree of bad press among network users, as they often see them as a trade-off for their convenience or ease of use of network services.

2. Evolution of firewall systems

A simple way of referring to firewall systems is in relation to the order of appearance; for example, first, second, and third generation firewall systems, respectively. The main difference lies in the level at which traffic filtering is done, that is, at the layer on which they act. Figure 1 shows this evolution, with respect to generations and performance layers, using the OSI reference model for the layers.

Figure 1. Evolution of first, second and third generation firewall systems regarding the analysis and action layers, using the OSI reference model.



Next Generation Firewall

Other names, such as last generation firewalls, or new generation firewalls (NGFW) are sometimes used in the commercial field, to refer to additional functions of third generation firewalls, such as support for encrypted traffic inspection with protocols such as TSL and SSH, active directory integration within the firewall, malware filtering, built-in network intrusion prevention, etc. Later, in the section on third generation firewalls, we will discuss some of these additional features.

In the first subsection of this section we will deal with the first generation firewall systems, based on the use of routers with packet filtering. In the following subsections, we will also discuss second generation firewalls (known as *stateful inspection firewalls*) and third generation firewalls (known as *application layer* or *layer 7 firewalls*, in relation to the OSI reference model). We will see representative examples, as well as the advantages and limitations for each type.

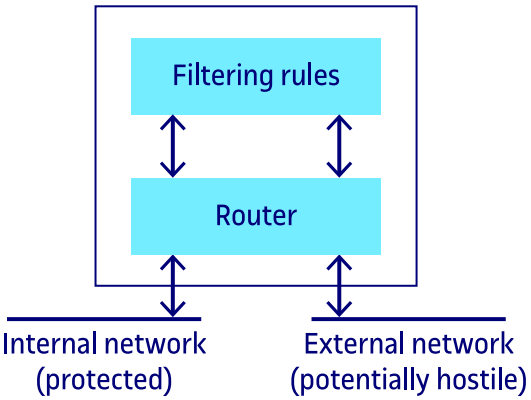
2.1. First generation: packet filtering at network level

As shown in figure 2, a router with packet filtering is a device that routes and inspects traffic at the network level (IP packets, for example). The router will decide whether or not to let the traffic pass according to filtering rules associated with a security policy.

Packet filtering firewalls

The router that filters traffic is also called a *screening router*.

Figure 2. First generation firewall: packet filtering firewall



Filtering rules are responsible for determining whether a packet is allowed to pass from the internal part of the network to the external part, and vice versa, by checking for legitimate data traffic between both parties.

Filtering rules associated with packet-filtering routers typically use information present in network packets traversing the firewall system. That is, they indicate which packets may or may not pass by looking at the headers of the associated protocols (for example, protocols such as ARP, IP or ICMP). This associated information can be:

- the source and destination addresses of the packets,
- the protocol type associated with the packets,
- the ports of origin and destination,
- the message type (at the network level),
- the contents of the packets (at the network level),
- the size of the packet,
- etc.

Note that although first-generation firewalls tend to use source and destination ports in their filtering rules in addition to protocol type, this does not mean that they perform a thorough inspection of the connections in the transport layer (layer 4, in relation to the OSI reference model). We will see later that second generation firewall systems, in addition to consulting this source and destination port information, can additionally check the connection, which we do not expect from a router acting as a firewall at the network level.

If the router uses a first-match filtering strategy, each packet arriving at the device will be compared with the filtering rules, starting at the beginning of the list until the first match is found.

If there is a match, then the action indicated by the rule is triggered (for example, deny the packet, accept it, or reroute it).

If no match is found, the *default policy* will be consulted to determine what action to take (for example, let the packet pass or discard it). If it is, for example, a deny by default policy, if there is no match with the packet, it is discarded.

Default policies

A deny by default policy is also called a *deny all* or *closed policy*, while a default acceptance policy is also called an *allow all* or *open policy*.

A *deny by default* policy tends to be more expensive to maintain, as it will require the administrator to explicitly state all services to be kept open (others will all be denied by default). A *default acceptance* policy seems simpler to administer, but it increases the risk of receiving attacks against the network, since it requires the administrator to explicitly indicate which packets to discard (the rest, by default, will be accepted in their entirety).

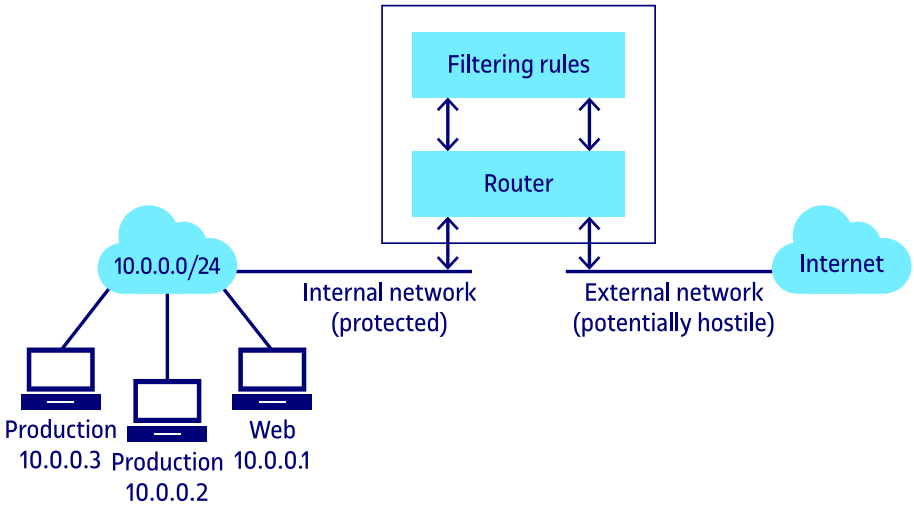
Most of the time, a default denial policy is chosen as a security measure. This strategy is sometimes called *the fail-safe security principle*.

A firewall system fulfils the **fail-safe** principle if it rejects an unexpected event, such as a packet for a new service.

Figure 3 shows an example diagram for the following security policy (very simplified, for ease of understanding):

- We assume a deny by default policy.
- All systems on the internal network (with network address 10.0.0.0/24) can access any service on the external network (Internet).
- External systems cannot connect to any internal system except the web server (computer with IP address 10.0.0.1).

Figure 3. Example of packet filtering with a first generation firewall.



The security policy configuration mentioned above, once applied by the router shown in figure 3, will allow all packets that have an internal network IP address as source and an external network (Internet) IP address as destina-

tion to pass through. It will also allow part of the traffic destined for the internal network, so that the computer with the 10.0.0.1 IP address can respond to requests from outside. Finally, since a deny by default policy is applied, the firewall will not allow packets that do not comply with the above rules to pass through. In addition, the rules corresponding to the response traffic must also be added to the router configuration, that is, rules representing the following two cases:

- allow traffic that enters the internal network and comes from web services (tcp source port 80)
- allow traffic that leaves the internal network and comes from the web server (10.0.0.1 source port tcp 80 and source IP address).

For the sake of simplicity, in this example we have not considered other web-related ports, such as port 443 related to TLS within HTTP (i.e., HTTPS), or the use of the Domain Name System (DNS) that would be needed in a real scenario.

So, the above example shows us that packet filtering is based on using the information available in the headers of a protocol such as IP, stateless, using data such as, for example, the source address, the destination address, source and destination ports, etc.

With this kind of data it is easy to specify rules of the type "accept all outgoing traffic intended for web services" (the way to do this is to consider tcp 80 as the destination port). On the contrary, it does not allow expressing rules of the type "accept all HTTP traffic only if it is not being used to download music". To be able to solve this case, it would be necessary to use a gateway-based firewall at the application layer, so that the firewall would analyze the traffic at the application layer level and detect whether it is being used to download music or not.

As stated before, each filtering rule has an action associated with it. This action determines what the firewall system should do with each packet that meets the conditions associated with the filtering rule. Examples of used actions are those that indicate that the packet can be accepted or rejected.

If a packet is rejected, there is the possibility of generating error messages. Typically, this will involve using ICMP-type traffic to notify the device that originated the packet of the firewall's decision to reject it. More specifically, ICMP type 3 (Destination Unreachable) messages, with codes like those shown in table 1. A second possibility is to silently reject the packet, without creating an ICMP message to inform the device that originated the packet.

ICMP Error Codes

Codes 9 and 10 were specially added to the ICMP specification for use with filtering systems. However, many firewall systems continue to use only codes 0 and 1, which were originally intended for other purposes.

Table 1. ICMP codes that a firewall can send when it rejects a packet

Type	Code	Description
3	0	Destination network unreachable
3	1	Destination host unreachable
3	9	Network administratively prohibited
3	10	Host administratively prohibited

Complementary readings

The following article provides more information about filter rule ordering and organization strategies for conflict resolution and configuration issues: **Garcia-Alfaro *et al.*** (2007). "Management of Exceptions on Access Control Policies", *22nd IFIP TC-11 International Information Security Conference, New Approaches for Security, Privacy and Trust in Complex Environments*, 97-108, Springer Nature. <http://dx.doi.org/10.1007/978-0-387-72367-9_9>

On the other hand, the following article provides information on troubleshooting strategies for configuring firewall systems: **Garcia-Alfaro *et al.*** (2008). "Complete Analysis of Configuration Rules to Guarantee Reliable Network Security Policies", *International Journal of Information Security*,7(2):103-122, April 2008, Springer Nature. <http://dx.doi.org/10.1007/s10207-007-0045-7>

Generating ICMP messages or not has advantages and disadvantages. On the one hand, sending the informative ICMP message means that the source can close the connection immediately, without needing to waste time and without trying to retransmit the rejected packets. But it adds the following problem: the ICMP message generated by the firewall system can be interpreted in different ways by the source equipment that receives it. It may even result in a performance penalty for the firewall system or provide information that potential attackers can later use against the system. So, for many authors, it is considered safer to not send informational ICMP messages and simply reject packets silently.

Finally, the order in which filtering rules are processed is a very important parameter to consider, as it can be used as a mechanism to resolve conflicts and configuration issues. For example, if two conflicting rules are found (for example, one that accepts the packet and one that rejects it, a *first-match* strategy will give priority to the first rule that matches the conditions of the treated packet .This first rule will therefore decide the action to be executed for the packet in question. In contrast, using a *last-match* strategy will give priority to the last rule that matches the conditions of the packet, perhaps with a completely different action from the rules that precede it in the order.

In general, it is the firewall system administrator who will decide the resolution strategies regarding the selection and order of the rules. These decisions can affect the efficiency of the firewall system, at the expense of simplifying its configuration. While a configuration based on first- or last-match strategies simplifies configuration, it can also have a negative impact on filtering efficiency, as it can place a burden on network packet processing. In contrast, grouping rules by types or tables, later applying jumps according to those types

or tables, can improve packet processing efficiency (although it can also hinder the expressiveness associated with firewall system configuration). All these decisions can be very relevant when it comes to protecting high-speed networks (gigabit and terabit networks).

2.1.1. Advantages and disadvantages of first generation firewalls

Building a firewall system using a router with packet filtering is really cheap, as it is usually done with hardware that is already available. In addition, it offers high performance in networks with a high traffic load. A packet filtering router example can be easily implemented from the routing systems of GNU/Linux-based operating systems, along with the associated Netfilter modules and system commands (iptables or nftables applications, depending on the version of Linux used).

Additionally, this technology allows the implementation of most of the necessary security policies.

Despite these advantages, network routers with packet filtering can have some shortcomings, such as:

- Many of the routers in use may be vulnerable to existing attacks (although most providers have appropriate update packages to address this). On the other hand, they usually do not have logging capabilities. This makes it difficult for the administrator to know if the router itself is being attacked.

- Its performance can deteriorate due to the use of excessively strict filtering and also make the device management process more difficult if this number of rules becomes very high.

- Filtering rules can become very complicated and sometimes cause possible distractions in their configuration to be exploited by an attacker to commit a security policy violation.

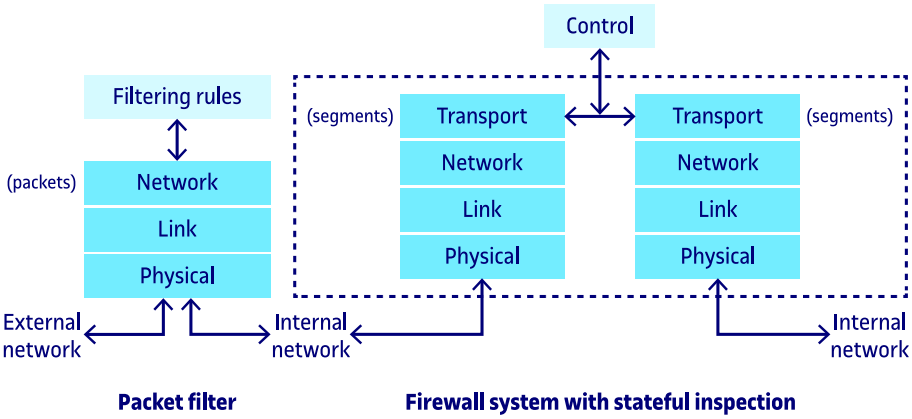
2.2. Second generation: stateful inspection at the transport level

Second generation firewall systems, also known as *gateways at the circuit level*, act as relays of traffic segments at the transport level. These devices filter content at the transport level (level 4) of the OSI layers reference model.

As we see in figure 4, these devices can supplement the packet processing done by a router with filtering rules, inspecting and deciding based on the states associated with TCP transport traffic segments, for example. That is, they can

hold network traffic until they get enough information about the end states of a communication and decide whether to allow or deny connections and associated data.

Figure 4. Second generation firewall, with stateful inspection, completing the packet filtering done by a first generation firewall



The **second generation firewall systems** are also known as ***stateful packet inspection firewalls***.

Stateful packet inspection firewall systems work at level 4 of the OSI reference model, that is, on the transport layer between two ends. Therefore, they can monitor and decide based on the connection with each of these two ends and decide whether to relay the associated segments or not. Apart from all the information already available at the network level, they will also be able to use information at the transport level, such as special flags, associated with the state of a connection at the TCP transport level, for example.

Note that although first generation firewalls can use the port number (located at the transport layer) in packet filtering rules, the systems do not perform stateful inspection of the connection.

2.2.1. Stateful protocol filtering

As we already anticipated in the introduction of this section, second generation firewall systems allow stateful filtering using information about the state of the connections or sessions associated with the traffic.

This information can be used to add greater richness to the filtering rules and allow packets to be accepted or rejected based on their membership in specific sessions or specific states of a protocol with information at the transport level. Packet processing performed by the firewall system will need to keep track of the state of the transactions associated with the packets or the behaviour of the traffic passing through them.

Differences between stateful and stateless protocol filtering

In packet filtering associated with a stateless protocol, such as IP-type network-level traffic, each packet can be processed independently of other packets (i.e., regardless of the order or priority of packets). In contrast, when filtering traffic associated with a stateful protocol, such as traffic segments at the transport layer of the UDP or TCP type, the firewall system will require storage and monitoring of the sequence of packets, the flags associated with connections, etc. In this way, the firewall can make the filtering decision once the packets are placed in the order defined by the logic of the protocol. This implies greater complexity in tracking connection states, the origin and destination of packets, use of ports associated with applications, etc. So, in the case of stateless protocol filtering it is usually called *static filtering*, while in the case of stateful protocol filtering it is usually called *dynamic filtering*.

Thanks to dynamic filtering, by performing stateful and previous connection inspection, second generation firewalls can establish much more compact filtering rules and reduce the size of the filter rule set (in relation to the static filtering of the first generation firewalls).

Assume, for example, the rule “accept packets received in response to a previous request, originating on the internal network”, along with the UDP packets listed in table 2.

Table 2. Example of UDP packets for the dynamic filtering example

1	Origin IP Origin port	230.0.113.1 43321	Destination IP Destination port	192.0.2.1 7
2	Origin IP Origin port	192.0.2.1 7	Destination IP Destination port	230.0.113.1 43321
3	Origin IP Origin port	192.0.2.1 7	Destination IP Destination port	230.0.113.1 34511

In the above example, we are considering an internal network with the network address 230.0.113.0/24. A firewall system configured with the rule described above will accept packets 1 and 2, indicated in table 2, but will reject packet 3, since it is a packet that does not correspond to any request originating within the same network. The firewall will consider this third packet as a violation of the network security policy.

2.2.2. Advantages and disadvantages of second generation firewalls

The dynamic filtering performed by second generation firewalls provides certain advantages over the static filtering of the first generation firewalls.

To begin with, and as we have seen with the example of the previous subsection (in relation to the UDP packets represented in table 2), this new type of filtering allows reducing the set of rules to implement the same policy as a first generation firewall, as there is no need to anticipate response packets. In addition, the new rules can also be easily implemented from devices running GNU/Linux and the Netfilter framework (iptables and nftables applications, also depending on the Linux version used).

A second advantage of second generation firewall systems is the expansion of the perimeter security concept. As we will see later, in the section on implementing architectures, second generation firewalls are usually used to allow the implementation of demilitarized zones (DMZ). In this case, the firewall can use specific security protocols to route from a protected area to an unprotected area. A specific example is the use of second generation firewalls together with the SOCKS (SOCKet Secure) protocol. This protocol consists of a client and a server. The server can run inside the firewall, while the client can run on internal computers in the DMZ. The firewall will evaluate connection requests and decide based on a security policy whether the connection should be allowed or not. If the connection is to be allowed, the functionality of the SOCKS protocol will be used to relay the traffic and allow input or output at the transport level.

Use of the SOCKS protocol by second generation firewall

The SOCKS protocol, defined in RFC 1928, is considered a de facto standard for implementing second generation firewalls with circuit-level gateways.

The combination of dynamic filtering, together with the relaying of UDP or TCP traffic through clients and servers of the SOCKS protocol, allows the definition of DMZ-type architectures, hiding IP addresses of protected equipment from potentially hostile traffic from the outside.

However, dynamic filtering is not perfect either. First of all, the rules defined in a firewall with dynamic filtering can be easily evaded, through IP spoofing attacks. Indeed, an attacker who succeeds in spoofing the IP address or the source port will be able to easily camouflage himself through the network and pass off his traffic as responses to previous requests, thereby fooling the connection tracker of the firewall. To avoid this, it will be necessary to think about more complex rules, which leads us to other drawbacks associated with this new type of filtering: the complexity of dynamic rules, the loss of efficiency in the treatment of traffic in congested networks, etc.

Complexity and loss of efficiency in dynamic filtering.

Dynamic filtering of stateful protocols increases the complexity of the rules, as well as a possible loss of efficiency. To be able to handle all possible cases associated with evasion or configuration errors, a firewall system with dynamic packet filtering will need additional memory resources to ensure connection and session tracking. If this is not possible, it will be difficult to guarantee correct processing of the state of the traffic flowing through the system. On the other hand, the increased complexity in packet processing, as well as the loss of efficiency can also open the possibility of suffering denial of service attacks. For example, adding dynamic rules to deal with potential IP spoofing attacks tends to lead to denial of service situations on firewall systems with limited storage and processing memory. A possible solution to these problems will be to add other defence devices within the system that needs to be protected. For example, the addition of attack and intrusion detection devices.

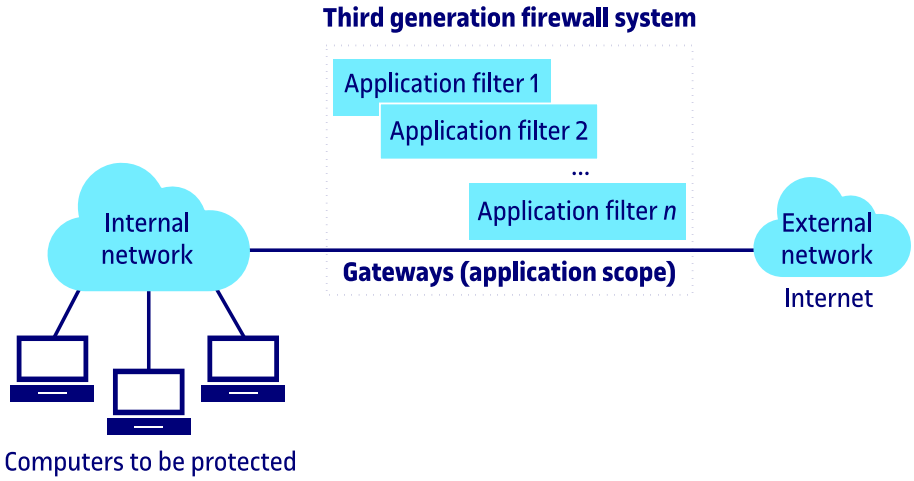
2.3. Third generation: application layer data processing

Third generation firewalls can enter to inspect (and modify) application layer information (level 7 of the OSI reference model). So they are built as application level gateways (proxy servers). In addition to being able to route packets at the network level or relay segments at the transport level, they act as gateways with access to application level data.

> A proxy server is responsible for making the requested connections with the outside, and when it receives a response, it is responsible for retransmitting it to the equipment that initiated the connection. Thus, the proxy server running on the gateway applies the security policy to decide whether to accept or reject the connection request.

In fact, layer 7 firewalls typically inspect, modify, or eliminate traffic using specialized filters for a predetermined set of applications. Figure 5 shows this idea. Each filter can be considered as an intermediary processor, capable of modifying content at the application layer level. Packets corresponding to an application protocol known by the firewall are directed to the specific filter, which will be responsible for inspecting and processing the received data. This allows not only to filter contents, but also to alter part of the data. For example, a layer 7 firewall can perform HTTP (Hypertext Transfer Protocol, RFC 8740) traffic inspection. As a result of this inspection, and depending on the security policy rules associated with the firewall, it may reject the traffic, or allow it to pass, but making modifications to the associated data (for example, the filter may remove or rewrite JavaScript code found in HTTP data, rewrite headers or data based on known URLs, etc.)

Figure 5. Third generation firewall performing filtering or content modification at the application level, by selecting specific filters for each type of application



As also shown in figure 5, the gateway found within the third generation firewall system separates the internal network (with computers to be protected) from the external (potentially hostile) network. But, unlike first and second

generation firewall systems, this separation is done at the application layer level. This allows additional protection in the area of users or application data, in parallel with the security analysis associated with each filter installed on the firewall. These filters can be updated frequently, adjusting the analysis to modifications and service updates in the application layer, without the need to make changes to the lower layers of the firewall system.

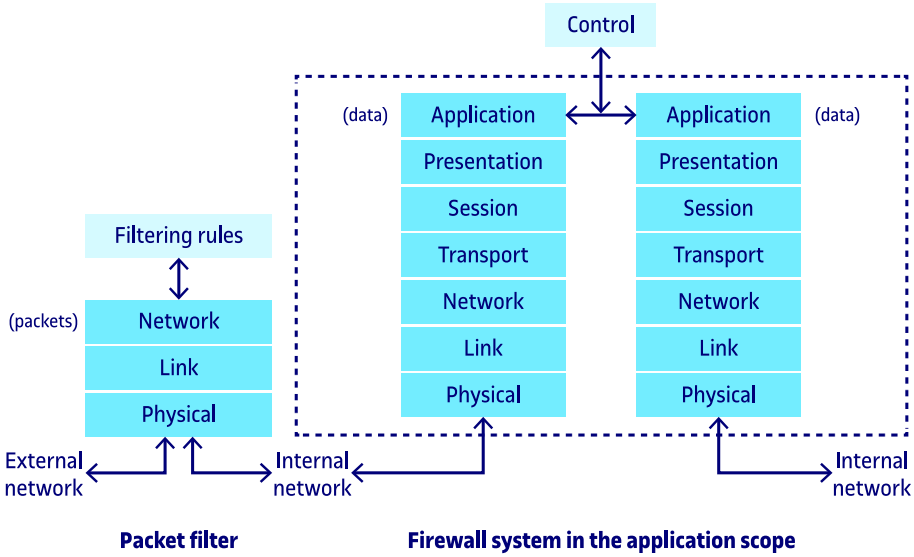
2.3.1. Advantages and disadvantages of third generation firewalls

Third generation firewalls offer advanced functionality with respect to packet filtering at the network level, or segment management at the transport level. They present a much higher range of possibilities. On the contrary, they also introduce a penalty to the traffic processing service, since they have to do a deep inspection of the data. In the case of congested networks, where the traffic load is high, the use of this kind of firewall can greatly penalize network performance and latencies.

A traditional way to solve this performance problem is to use cache systems that keep a local copy of the previous data received by the firewall, to be reused on subsequent connections (if this is possible). Despite this, the dynamism of current services, in which encrypted content changes continuously, also hinders the improvements provided by this cache-based solution.

A second way to deal with performance issues is to combine third generation firewalls with packet (first generation) or transport (second generation) pre-filtering. Thus, the simplest cases can be handled initially with traditional packet filtering, or stateful inspection if necessary, before moving on to application-level data inspection or modification. Figure 6 shows this last idea and illustrates the combination of packet filtering only, with application level analysis. This way of combining both systems, in addition to helping to reduce the loss of performance, also helps to provide flexibility at the configuration level.

Figure 6. Packet filtering and data gateway at the application level



The use of gateways within third generation firewall systems, with specific filters at the application level, provides other benefits. For example, it can enable effective identification as well as subsequent filtering of misuse of applications prohibited by the security policy. This last example is used to detect users trying to escape the ban of online gaming services or the use of P2P applications to download illegal material. Although this can also be done with traditional filtering at the network or transport level, this new identification can be based on the analysis of the data processed directly at the gateway level. That is, instead of making an identification according to ports or protocols indicated at lower levels, an identification of improper uses will be made based on the filters of the application level.

Another benefit of having the gateway at the application level is that the application protocol can also differentiate between specific situations when filtering. For example, the firewall can be configured to allow the use of P2P applications only for downloading operating systems or service updates, but continue to filter the download of illegal material (copyrighted films or music) within the same traffic flow. This feature would not be possible solely with network or transport level filters within first or second generation firewalls.

On the other hand, third generation firewall systems, despite offering more control over monitored services, still present certain drawbacks. A first drawback to highlight is the need to configure a filter for each service to be monitored. Creating filters for traditional services is not a problem, i.e., services with known flows for tracking sessions like TELNET, FTP, HTTP, etc. However, creating filters for little-known services or proprietary protocols without documentation can be a difficult problem to solve. The creation of these filters requires very detailed knowledge at the level of specifications. This tends to make it difficult to create specific filters for old protocols, for example, indus-

trial protocols, without standardized specifications. Email-related traffic can also lead to problems and produce spam filtering, or the specific removal of macros or executables within messages, incorrectly.

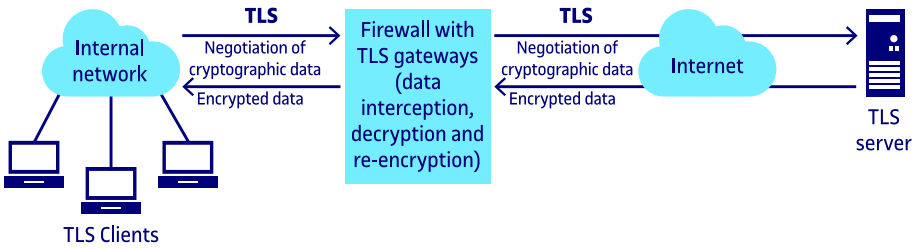
In fact, many of these drawbacks are also found in intruder detection systems, since the same concept of filters for the identification of improper uses and modification of known flows at the application level is the basis of these cyber defence tools.

A final drawback with analyzing data at the application layer is that this data is often encrypted because it is using transport layer security. The solutions to this kind of problem are varied and are beyond the scope of this material. Even so, as already mentioned briefly at the beginning of this section, it is one of the functions that are usually used as a commercial claim for Next Generation Firewalls (or NGFW).

These next generation firewalls incorporate new functionalities over gateways at the application layer of third generation firewalls, to perform (among others) threat detection, integrated network intrusion prevention, and malware filtering at system level and, in the case which concerns us here, encrypted traffic inspection.

This last functionality is usually used to carry out TLS inspection. Both TLS traffic inspection and other cases (such as SSH traffic inspection or similar) assume the incorporation of gateways at the application level, as well as the necessary filters to perform interception tasks, certificate negotiation, key and cryptographic data negotiation, online decryption, inspection and re-encryption. Figure 7 shows a simplified example of this technique, using a TLS gateway within the firewall system.

Figure 7. TLS traffic inspection by a next generation firewall.



Inspection of encrypted traffic by next generation firewall systems is a controversial practice. Apart from offering and using common techniques in interception attacks (man-in-the-middle), as well as possible violations of end-to-end encryption (responsible for protecting communications from origin to the final recipient), it also brings ethical problems when used to inspect corporate traffic, since the decryption of private data will not always be viewed favourably by the users of the organization, which leads to a possible intrusion into their privacy. Even so, it is a functionality increasingly demanded to solve the problem of evasion of hidden attacks in encrypted traffic (both to escape the filtering of a firewall system, and for the detection or prevention of intruders by detection systems).

Recommended readings

The following articles (both available online) provide more information on inspecting TLS-encrypted traffic and the potential consequences in the area of security or malpractice.

O'Neill *et al.* (2017). "TLS Inspection: How often and who cares". *IEEE Internet Computing*, IEEE Computer Society. <https://doi.org/10.1109/MIC.2017.265102655>

Durumeric *et al.* (2017). "The Security Impact of HTTPS Interception", NDSS Symposium. <http://dx.doi.org/10.14722/ndss.2017.23456>

3. Implementation of perimeter security through firewall systems

An effective implementation of the security policies associated with a system will largely depend on the architecture of the system to be protected. It is also very important to bear in mind the type of firewall systems that will implement this security policy, the location on the network or networks of the system, the combination with other equipment, etc.

Remember that, in general, the goal of implementing a security policy through firewall systems is to protect internal networks from the outside. Usually, this outside part tends to be the Internet. However, in other cases, firewall systems can be used to separate parts of the same internal network, such as the workstations of an industrial system or a test laboratory.

Numerous architectures and types of strategies exist in the firewall system literature. Although there is no sufficiently agreed-upon classification, we can simplify and divide the main filtering architectures or strategies into the following two types:

- single point architectures
- perimeter network architectures

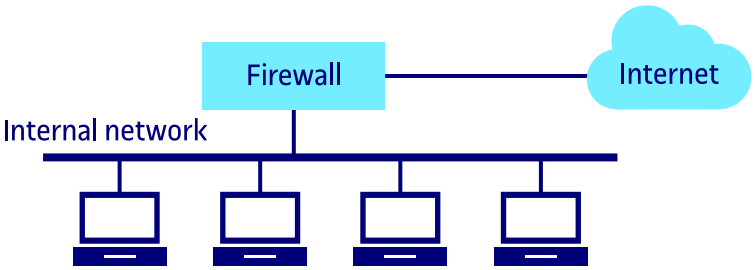
Next, we will present these two types of architecture in more detail.

3.1. Single point architectures

This first architecture is the simplest to implement. It consists of separating the network that we want to protect from the outside with a single protection device, as shown in figure 8. The device represents a single point of configuration. This makes the resulting filtering system simpler to implement and administer. At the same time, it also makes this single configuration item the critical point of the system. If an attacker manages to compromise any of the services behind this single configuration point, the computers associated with the protected system can be attacked without restriction.

The device labelled as a firewall in figure 8 can be based simply on a router with packet filtering (i.e., a first generation firewall system) or a device with more filtering capabilities, along with gateways inside to establish accepted communications (i.e., a second or third generation firewall system). From an architectural point of view, we will label this second case (second or third generation firewalls with gateways inside) with the concept of bastion host.

Figure 8. Single point architecture



A **bastion host** is a highly protected system prepared to withstand attacks from a hostile place (in this case, the Internet) and which usually acts as a point of communication between the inside and the outside of a network.

Bastion host

The name *bastion host* comes from the heavily protected walls that separated medieval castles from the outside.

The bastion host is usually configured with two or more network interfaces, with the routing service disabled by default. Thus, traffic from one end of the network (the hostile side) will not be routed to the other side (the protected side) by default. Only if explicitly accepted, a gateway installed in the bastion host will be responsible for making the connections on behalf of these two parties. It will also allow the redirection of traffic to other networks, to carry out a more detailed analysis.

However, the use of these bastion hosts with second and third generation firewalls including intermediate gateways leads to a loss of efficiency as the main drawback, especially if the traffic that travels through it is congested. To solve this problem, it will be more efficient to diversify and introduce multiple computers, combining routing and filtering at different levels (network, transport or application), as we will see in the next subsection.

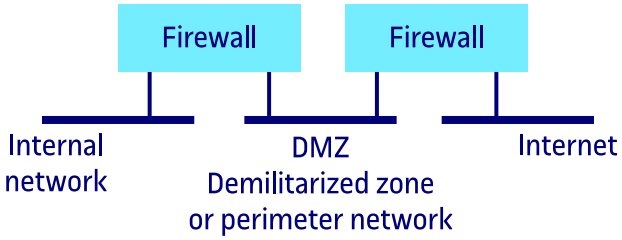
3.2. Architectures with perimeter networks

As already indicated above, regarding figure 8 of the previous subsection, we will differentiate with the firewall label the case of a first generation system, that is, a router with packet filtering, with respect to systems of second and third generation, which will be identified in the figures in this subsection with the bastion host label. With this proposed notation, and only from an architectural point of view, we will try to differentiate the use of filtering options with economical and efficient firewalls (with a traffic analysis at the network layer level), with more expensive schemes of filtering but offering more protection, assuming in addition the required gateway services to establish the accepted communications.

A first way to make it more flexible and add more security to the single-point architectures discussed earlier is the use of architectures with perimeter networks. In this case, we will add a subnet between the internal and external

network to act as a barrier against possible attacks and intrusions. This perimeter network (or the set of multiple perimeter networks) is also known as a demilitarized zone(s) (DMZ). Figure 9 shows a very simplified example.

Figure 9. Incorporation of a demilitarized zone or perimeter network



DMZ, perimeter networks and bastion host

The terms DMZ and perimeter network are used synonymously by many authors.

In general, a DMZ is considered to be the set of perimeter networks. In other words, a DMZ can consist of one or more perimeter networks.

Within each perimeter network, we will find bastion host (one or more), which usually refer to the devices that are continuously exposed to attacks.

Bastion hosts can provide public services of the organization (web, email, DNS, etc.), in addition to protection services (among others, traffic filtering at the transport or application level). That is why, in this section, we will use the concept of bastion host to refer to devices that provide second and third generation firewall services, to differentiate them from the input and output firewalls of a DMZ, generally implemented with first generation firewalls (i.e., routers with packet filtering at the network level).

Each perimeter network usually has one or more bastion hosts inside. These bastion hosts can offer, among other things, the filtering and gateway services discussed above. This will provide, in addition to a higher level of security, complete separation from the internal network. In addition, the bastion computers tend to give (or establish) access to internal network services that must be accessible from the outside.

If an attacker manages to bypass the security of the first firewall (or external firewall) and enter the perimeter network, he will not be able to immediately attack the internal network equipment, since they are protected by the second firewall (or internal firewall).

In figure 10 we can see in more detail a first perimeter network architecture. To simplify, we differentiate between two first generation firewall systems (outer and internal firewalls), as well as a bastion host (with a second or third generation firewall system inside). So let's assume that the external and internal firewalls are routers with packet filtering, and that the bastion host is configured with a minimum of two network interfaces, with routing disabled by default, but with gateways in charge of filtering tasks at the transport or application level, as well as the establishment of the necessary connections. Although the example in figure 10 shows only one bastion host, there could be more, as we will see later.

Figure 10. Simplified example of an architecture with perimeter network

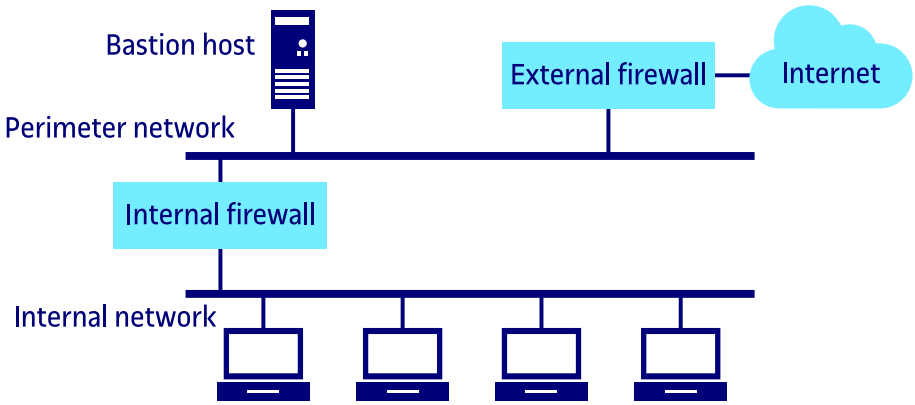


Figure based on the work of Zwicky, Cooper and Chapman (2000).

Note that the function of the two first generation firewalls shown in figure 10 is different. On the one hand, the firewall labelled as an internal firewall protects the internal network from the external network, but also from the perimeter network. We can see it as a router with packet filtering, to eliminate dangerous traffic (both input and output) to the internal network, from the outside world. That is, the internal firewall also controls the traffic between the internal network and the bastion host, thus ensuring that the traffic between the internal network equipment and the services of the bastion host is extremely limited, to prevent the commitment of the bastion host entailing a possibility of attacking the teams of the internal network. In addition, the bastion host can also offer traffic filtering at the transport and application level.

On the other hand, the external firewall protects both the internal network and the perimeter network. Again, we can see it as a router with packet filtering, but with less restrictive filtering rules. In fact, their rules will be specially designed to protect the bastion host from the outside. This external firewall may even be controlled by an external organization (for example, an Internet service provider).

Note that the initial example in figure 10 could also be configured more compactly, using a single packet filtering at the network layer, thus placing a single first generation firewall, routing and filtering tasks between the inner and outer part of the system, as we show in figure 11.

Figure 11. Example of architecture with a perimeter network with a single point of input and output.

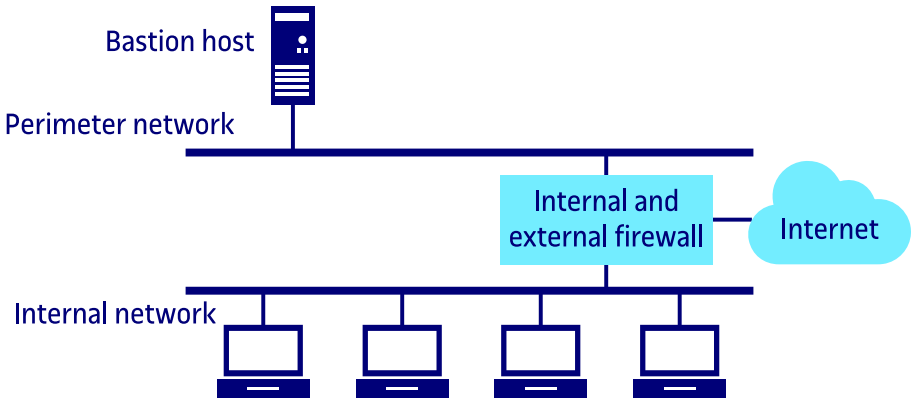


Figure based on the work of Zwicky, Cooper and Chapman (2000).

In any case, these two architectures with a single perimeter network allow us to see how different types of firewall systems can be combined. The main idea is to establish a first barrier of entry from a simple firewall, later combined with one or more system(s) configured with more advanced filtering services, together with the necessary gateways to establish the final communications, in case the traffic is accepted. These first two architectures shown in figures 10 and 11 aim to ensure a first level of protection, as well as addressing the performance issues associated with each type of filtering technology.

The architectures shown in figures 10 and 11 can be further generalized, for example, by splitting the initial perimeter network with other perimeter (sub)networks. Figure 12 shows a first example of this idea, increasing the number of perimeter networks and bastion equipment by one unit. A typical rationale for this architecture shown in figure 12 is to add redundancy and diversification to the services offered by the initial bastion host (not only the protection services at the level of filtering or data processing, but also the own services hosted on the bastion host for administrative tasks, for example). Again, this architecture will increase the security or efficiency levels of internal network equipment already shown in previous architectures, but with higher configuration and management costs.

Figure 12. Example of architecture with two perimeter networks.

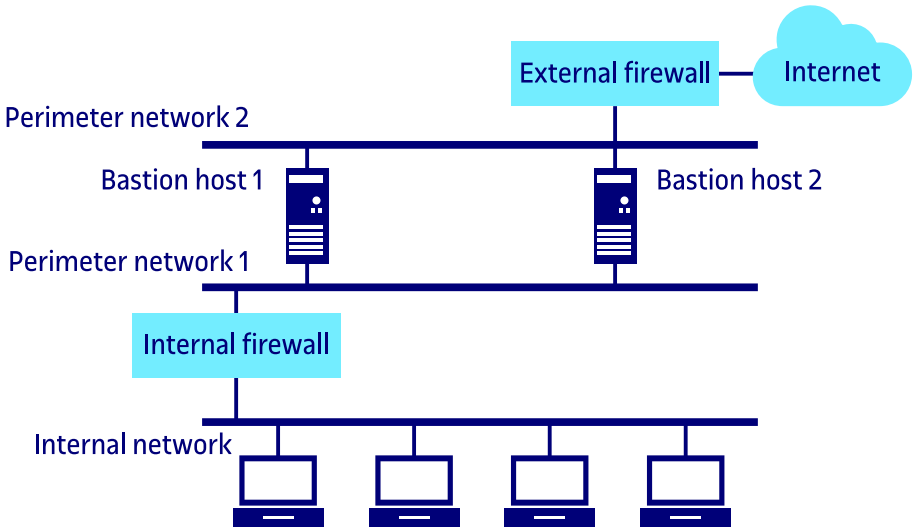


Figure based on the work of Zwicky, Cooper and Chapman (2000).

By evolving the previous architectures, we can reach much more complex situations, such as the architecture shown in figure 13. In this case, we find multiple perimeter networks further increasing the final redundancy of the system and also adding independent accesses to the Internet. This last configuration can be used as a measure of traffic separation between multiple perimeter networks, with different degrees of confidentiality; or to separate incoming traffic on the organization's servers from outgoing traffic of the organization itself. As in the previous cases, despite offering a much higher degree of protection, as well as improving the efficiency in the treatment of traffic, it will have as its main drawback a much more complex administration and configuration than the previous architectures. This can again lead to configuration errors and leave vulnerable spaces unprotected by mistake. It will also require a broader security treatment using other tools, such as monitoring and intrusion detection tools.

Figure 13. Example of a complex architecture with multiple perimeter networks

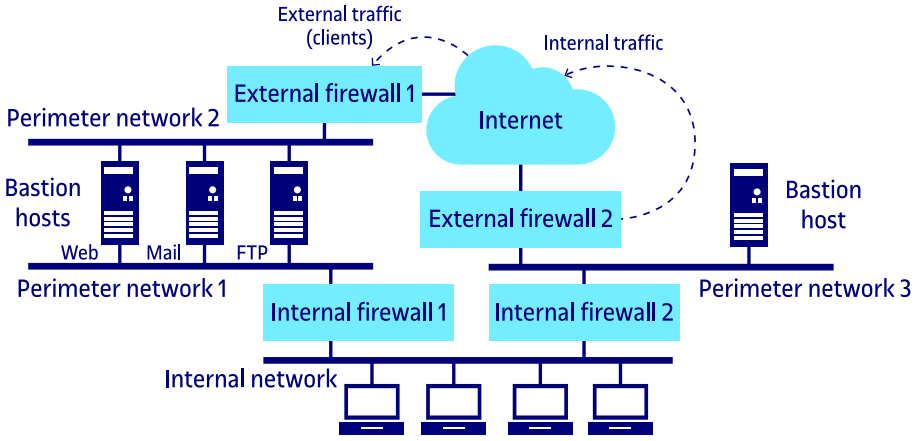


Figure based on the work of Zwicky, Cooper and Chapman (2000).

Summary

When a system is connected to a computer network, it is exposed to a set of threats that are always present. Furthermore, these systems are highly likely to have vulnerabilities which increase the likelihood of these threats taking place.

Firewall systems focus security decisions on a single point that is located where the greatest vulnerabilities exist and deny any connection that is not expressly allowed.

Through a packet filter configuration scenario in simple firewall systems, the decisions of a security policy defined by the organization can be technologically applied.

It is also possible to build firewall systems using proxy or gateway technologies, so that all received traffic can be interpreted at higher stack levels.

So, a firewall system is a control barrier that will keep the network protected from all unauthorized access and will act as a central point of control, simplifying the administration tasks.

On the other hand, due to the fact that they are located at a collision point, firewall systems offer other interesting security functions, such as the monitoring of network connections; content analysis (to search for viruses, for example); perform additional authentication checks; construction of virtual private networks; etc. They can also perform functions not directly related to network security, such as network address translation (NAT), network service management, bandwidth control, etc.

Finally, we must bear in mind that firewall systems are only prevention mechanisms and that they are not a single solution to solve all the security problems of a network connected to the Internet. These systems will never be able to protect the network from attacks that happen inside it and an external attacker may be helped by an internal (legitimate) user to collaborate in the attacks. Neither will they be able to prevent attacks against services with global access (in which anyone can access from anywhere) nor will they be able to protect the network against the transfer of malicious applications (viruses, worms, etc.). It would be impractical to use a device dedicated to analyzing all the traffic that circulates through it. This is why additional protection mechanisms are needed, such as intruder detection systems.

Glossary

Attack attack on the security of a system resulting from an intentional and deliberate act that violates the security policy of this system.

Bastion equipment see *Bastion host*.

Bastion host computer system that has been strongly protected to withstand attacks from a hostile location.

Demilitarized zone (DMZ) within a network protected by a firewall, an area separated from the public servers by a second firewall.

DMZ see *Demilitarized zone*.

DNS see *Domain Name System*.

Domain Name System hierarchical and distributed naming system that allows domain names to be associated with IP addresses.
acronym **DNS**

Dual-homed machine equipment with at least two network interfaces, each one associated with a network, that can act as a router between networks.

Firewall prevention element that will perform an access control to separate our network from outside (potentially hostile) equipment.

Gateway at circuit level device that acts as a gateway at the level of the transport layer between two ends. It establishes a connection with each one and relays data between the two connections.

ICMP see *Internet Control Message Protocol*.

Internet Control Message Protocol control protocol, mainly for sending TCP/IP error messages.
acronym **ICMP**

Internet Protocol protocol for interconnecting networks.
acronym **IP**

IP see *Internet Protocol*.

IP Address address that uses the IP protocol.

Perimeter security security based only on the integration of firewall systems and other prevention mechanisms into the network.

Proxy server software that will be responsible for making the requested connections with the outside and relaying them to the equipment that initiated the connection.

Router with packet filtering network device that routes TCP/IP traffic based on a series of filtering rules that decide which packets are routed through it and which ones are discarded.

Security policy set of rules and practices that define and regulate the security services of an organitzation or system with the purpose of protecting its critical and sensitive resources. In other words, it is a statement of what is permissible and what is not.

TCP see *Transmission Control Protocol*.

Threat potential violation of security based on circumstances, capabilities, actions or events that may cause a breach of security or cause damage to the system.

Transmission Control Protocol TCP/IP (end-to-end) transport protocol.
acronym **TCP**

UDP see *User Datagram Protocol*.

User Datagram Protocol TCP/IP (end-to-end) transport protocol.
acronym **UDP**

Bibliography

Avolio, F. (1999). "Firewalls and Internet Security, the Second Hundred (Internet) Years". *The Internet Protocol Journal* (vol. 2, no. 2).

Cheswick, W.; Bellovin, S.; Rubin, A. (2003). *Firewalls and Internet Security: Repelling the Wily Hacker* (2nd ed.). Boston, Massachusetts: Addison-Wesley Professional Computing.

Fraser, B. (1997). "Site Security Handbook". *RFC 2196, IETF. The Internet Society*.

Garcia-Alfaro, J. (2004). "Mecanismos de prevención". In: Herrera Joancomartí (coord.); Garcia-Alfaro; Perramón. *Seguridad en redes de computadores*. Barcelona: Fundació Universitat Oberta de Catalunya, 287 pp.

Microsoft (2010). *Windows Firewall with Advanced Security Getting Started Guide* (online). Microsoft TechNet Library.

Navarro Arribas, G. (2012). "Sistemes tallafoc". In: Herrera Joancomartí (coord.); Borrell, Garcia-Alfaro, Martínez, Navarro, Pérez, Perramón, Rifà, Robles. *Seguretat en xarxes*. Barcelona: Fundació Universitat Oberta de Catalunya.

Rash, M. (2007). *Linux Firewalls: Attack Detection and Response with iptables, psad, and fwsnort*. San Francisco, California: No Starch Press.

Shirey, R. (2000). "Internet Security Glossary". RFC 2828, IETF. The Internet Society.

Zwicky, E.; Cooper, S.; Chapman, D. (2000). *Building Internet Firewalls* (2nd ed.). Sebastopol, California: O'Reilly Media.