Firewall Technologies

Lecturer: Nguyen Hieu Minh
Outline

- Firewall overview
- Traffic control and the OSI reference model
- Firewall categories
- Firewall design
1. Firewall Overview

- Firewall technologies have undergone substantial changes since their entry into the marketplace in the early 1990s.
- These first firewalls were simple packet-filtering devices.
- Since those days, firewalls have become much more sophisticated in their filtering features, adding such capabilities as stateful filtering, VPNs, IDS, multicast routing, connection authentication, DHCP services, and many others.
One of the driving forces of these enhancements, besides vendor competition, was the explosion of Internet usage in the mid- to late 1990s.

The need to protect a company's assets, firewalls have become a common technology for not only enterprise companies, but also small businesses and personal computers that have Internet access.
Definition of a Firewall

- People use many descriptions when defining a firewall.
  - Its first use had to do not with network security, but with controlling actual fires.
- Of course, when we talk about network security, the term firewall means something different, but the original essence is carried over:
  - It is used to protect your network from malicious people and to stop their illicit actions at defined boundary points.
(cont.)

- Basically, a firewall is a device or systems that control the flow of traffic between different areas of your network.

- Notice something important about this definition:
  - The definition can include one or more devices.
    - a small office/home office.
    - an enterprise network
(cont.)

- Many people assume that firewalls are used to protect assets from external threats (from the Internet, TCP/IP).
- However, most malicious network threats and attacks occur, interestingly enough, within the interior of your network (have more than one protocol running).
  - A comprehensive firewall solution must be capable of dealing not only with both internal and external threats, but also with multiple protocols.
Firewall Protection

- Firewall systems can perform many functions and offer many solutions.
  - However, one of its primary purposes is to control access to resources.
- You can use many methods to perform this task.
Securing All Network Devices
In this example, firewall software is installed on each PC and file server, and is configured to allow only certain types of traffic to enter or leave the machine.

- This works well in a small office with only a handful of devices that need to be secured.
- In a network with tens of thousands of devices, this becomes problematic.
Securing All Network Devices
In this example, because the firewall solution is implemented in one device, it becomes much easier to manage security policies and their implementation.

With a single device, it becomes easier to restrict traffic entering and leaving the network:

- You set up the policies only once instead of on all the internal devices.
- This also reduces the total cost of the solution.
2. Controlling Traffic and the OSI Reference Model

- A good place to start is to review the Open System Interconnection (OSI) reference model.

- Using the OSI reference model will help you understand how firewalls process traffic.
Firewalls and the OSI Reference Model

<table>
<thead>
<tr>
<th>Layer 7</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer 6</td>
<td>Presentation</td>
</tr>
<tr>
<td>Layer 5</td>
<td>Session</td>
</tr>
<tr>
<td>Layer 4</td>
<td>Transport</td>
</tr>
<tr>
<td>Layer 3</td>
<td>Network</td>
</tr>
<tr>
<td>Layer 2</td>
<td>Data Link</td>
</tr>
<tr>
<td>Layer 1</td>
<td>Physical</td>
</tr>
</tbody>
</table>

Diagram:

- Layer 7: Application
- Layer 6: Presentation
- Layer 5: Session
- Layer 4: Transport
- Layer 3: Network
- Layer 2: Data Link
- Layer 1: Physical

Firewall Layers

9/1/2012
A firewall system can operate at five of the seven layers of the OSI reference model.

However, most firewall systems operate at only four layers: the data link, network, transport, and application layers.
The more layers that a firewall product or solution can cover, the more thorough and effective it can be in restricting access to and from devices.

For example, a firewall that operates at only Layers 3 or 4 can filter only on IP protocol information, IP addresses, and TCP or UDP port numbers; it cannot filter on application information such as user authentication or commands that a user enters.
3. Firewall Categories

- A firewall system can be composed of many different devices and components.
  - One of those components is the filtering of traffic, which is what most people commonly call a firewall.
Filtering firewalls come in many different flavors, including the following:

- Packet-filtering firewalls
- Stateful firewalls
- Application gateway firewalls
- Address-translation firewalls
- Host-based (server and personal) firewalls
- Hybrid firewalls
3.1. Packet-Filtering Firewalls

- The simplest form of a firewall is a packet-filtering firewall.
- A packet-filtering firewall is typically a router that has the capability to filter on some of the contents of packets.
- The information that the packet-filtering firewall can examine includes Layer 3 and sometimes Layer 4 information.
Packet Filtering Firewalls and the OSI Reference Model
Because TCP/IP is the de facto standard of communications protocols in today's networks, most packet-filtering firewalls support at least this protocol.

However, packet-filtering firewalls can support other protocols as well, including IPX, AppleTalk, DECnet, and Layer 2 MAC address and bridging information.
Filtering Actions

- When implementing packet filtering, packet-filtering rules are defined on the firewall.
  - These rules are used to match on packet contents to determine which traffic is allowed and which is denied.

- When denying traffic, two actions can be taken: *notify the sender of traffic that its data was dropped* or *discard the data without any notification*. 
Filtering Information

- A packet-filtering firewall can filter on the following types of information:
  - Source and destination Layer 3 address
  - Layer 3 protocol information
  - Layer 4 protocol information
  - Interface of sent or received traffic
## TCP/IP Packet Filtering Information

<table>
<thead>
<tr>
<th>Layer</th>
<th>Filtered Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>IP addresses</td>
</tr>
<tr>
<td>3</td>
<td>TCP/IP protocols, such as IP, ICMP, OSPF, TCP, UDP, and others</td>
</tr>
<tr>
<td>4</td>
<td>IP precedence (type of service [ToS]) information</td>
</tr>
<tr>
<td>4</td>
<td>TCP and UDP port numbers</td>
</tr>
<tr>
<td>4</td>
<td>TCP control flags, such as SYN, ACK, FIN, PSH, RST, and others</td>
</tr>
</tbody>
</table>
Packet-Filtering Firewall Example
## Packet-Filtering Table

<table>
<thead>
<tr>
<th>Rule</th>
<th>Source address</th>
<th>Des address</th>
<th>IP protocol</th>
<th>Ip protocol inf</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Any</td>
<td>200.1.1.2</td>
<td>TCP</td>
<td>Port 80</td>
<td>Allow</td>
</tr>
<tr>
<td>2</td>
<td>Any</td>
<td>200.1.1.3</td>
<td>UDP</td>
<td>Port 53</td>
<td>Allow</td>
</tr>
<tr>
<td>3</td>
<td>Any</td>
<td>200.1.1.4</td>
<td>TCP</td>
<td>Port 25</td>
<td>Allow</td>
</tr>
<tr>
<td>4</td>
<td>Any</td>
<td>Any other address</td>
<td>Any</td>
<td>Any</td>
<td>Drop</td>
</tr>
</tbody>
</table>

9/1/2012
In this example, rule 1 states that if traffic from any device on the Internet is sent to TCP port 80 of 200.1.1.2, the packet-filtering firewall should allow it.

Likewise, if any traffic is sent to UDP port 53 of 200.1.1.3 or TCP port 25 of 200.1.1.4, the traffic should be allowed.

Any other type of traffic should be dropped.
• It is important to point out that if you omit rule 4, you might have issues with a packet-filtering firewall.

• A packet-filtering firewall will make one of two assumptions:
  • If there is no match in the rule set, allow the traffic.
  • If there is no match in the rule set, drop the traffic.
For example

- Assume that you have a packet-filtering firewall that used the first process.
- In this example, if you omitted rule 4 in Table, if there were no matches in rules 1 through 3, all other traffic would be permitted.
(cont.)

- If your packet-filtering firewall uses the second process,
- If you omitted rule 4 in Table, any traffic that did not match the first three rules would be dropped.
Advantages of Packet-Filtering Firewalls

- Packet-filtering firewalls have two main advantages:
  - They can process packets at very fast speeds.
  - They easily can match on most fields in Layer 3 packets and Layer 4 segment headers, providing a lot of flexibility in implementing security policies.
(cont.)

- Because packet-filtering firewalls examine only Layer 3 and/or Layer 4 information, many routing products support this type of filtering.
- Because routers are typically at the perimeter of your network, providing WAN and MAN access, you can use packet filtering to provide an additional layer of security.
Limitations of Packet-Filtering Firewalls

- Despite their advantages, packet-filtering firewalls have these disadvantages:
  - They can be complex to configure.
  - They cannot prevent application-layer attacks.
  - They are susceptible to certain types of TCP/IP protocol attacks.
  - They do not support user authentication of connections.
  - They have limited logging capabilities.
Uses for Packet-Filtering Firewalls

- Because of these limitations, packet-filtering firewalls typically are used in the following areas:
  - As a first line of defense (perimeter router)
  - When security policies can be implemented completely in a packet filter and authentication is not an issue
  - In SOHO networks that require minimal security and are concerned about cost
3.2. Stateful Firewalls

- Unlike packet-filtering firewalls, stateful firewalls keep track of the state of a connection: whether the connection is in an initiation, data transfer, or termination state.

- This is useful when you want to deny the initiation of connections from external devices, but allow your users to establish connections to these devices and permit the responses to come back through the stateful firewall.
Many security people disagree on what layer of the OSI reference model stateful firewalls function at: Layers 3 and 4 (transport), or Layers 3, 4, and 5 (session).
From a transport layer perspective, the stateful firewall examines information in the headers of Layer 3 packets and Layer 4 segments.

For example, it looks at the TCP header for SYN, RST, ACK, FIN, and other control codes to determine the state of the connection.
(cont.)

- However, because the session layer establishes and tears down the connection—the transport layer handles the actual mechanics of the connection—some say that stateful firewalls operate at Layer 5.
Stateful Firewalls and the OSI Reference Model
3.3. Problems with Packet-Filtering Firewalls

- This section and the next one examine one of the issues that packet-filtering firewalls have with traffic and how stateful firewalls can deal with it.
Packet-Filtering Firewall Example—Initiating Connections
In the figure, the packet-filtering firewall has a rule placed on its inbound interface from the Internet stating that any external traffic sent to 200.1.1.10 (a user's PC) is denied.

As shown in Figure, when 170.1.1.1 tries to access 200.1.1.10, the packet-filtering firewall drops the traffic, as it is supposed to do.
However, what happens if someone inside the network, such as 200.1.1.10, tries to access this external device (170.1.1.1)?

Assume that this is an HTTP request to 170.1.1.1, which has a web server running on it.

HTTP uses TCP, and TCP goes through a three-way handshake to establish a connection before data is transferred: SYN, SYN/ACK, and ACK.
(cont.)

- Initially, 200.1.1.10 sends a SYN to establish a connection. With TCP (and UDP), a source port number is chosen that is greater than 1,023, which represents this specific connection.

- The destination is port 80, telling 170.1.1.1 that this is an HTTP request for web services.
(cont.)

- As the packet-filtering firewall receives the traffic on its internal interface, it checks to see if the traffic for 200.1.1.10 is allowed to leave the network.

  - In this case, no filtering rules prevent this, so traffic for 200.1.1.10 traffic is sent to the 170.1.1.1.
(cont.)

170.1.1.1 now responds back to the TCP SYN message of 200.1.1.10 with a SYN/ACK (the second step in the three-way handshake), as shown in Figure.

However, when the packet-filtering firewall examines the packet, it determines that because the destination is 200.1.1.10, the packet should be dropped, according to its packet-filtering rules.
(cont.)

- Therefore, the connection cannot be set up to the external web server, denying the internal user's web access.
Opening Ports

- You can solve this problem with packet-filtering firewalls in two ways:
  - Open destination ports greater than 1023 as traffic comes back to the source.
  - Examine the TCP control bits to determine whether this is returning traffic.
Take a look at the first solution

- In this situation, the source originally opened a source port greater than 1023, such as 10,000, and used a destination port of 80 for HTTP.
- Therefore, to allow the traffic to return from 170.1.1.1, the packet-filtering firewall needs a rule that will allow port 10,000.
(cont.)

- Of course, the problem with this is that the source can use any source port number greater than 1023: Whichever one is free and is chosen by the operating system is the one assigned.

- Therefore, you would have to allow all ports greater than 1023 to allow the returning traffic to 200.1.1.10, as shown in Figure.
Packet-Filtering Firewall Example—Opening Ports

Filter Rules Applied Inbound

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>Protocol Information</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any</td>
<td>Any</td>
<td>Destination Port &gt; 1,023</td>
<td>Permit</td>
</tr>
<tr>
<td>Any</td>
<td>200.1.1.10</td>
<td>Anything Else</td>
<td>Deny</td>
</tr>
</tbody>
</table>

Diagram:
- Two IP addresses: 200.1.1.10 and 200.1.1.11
- A Workgroup Switch
- A Packet-Filtering Firewall
- A Web Server at 200.1.1.2
- An Outside User at 170.1.1.1
- Connections between devices showing IP packets.
CAUTION

- Opening ports greater than 1023 is not a recommended practice to allow returning traffic from an originating connection: You are creating a huge security hole in your firewall that will open your internal devices to all kinds of attacks.
Examining TCP Control Bits

- The second approach is to examine transport layer information about the connection to determine whether it is part of an existing connection and, if so, allow the returning traffic back to 200.1.1.1.
With TCP, this can be done by examining the control flags in the TCP segment header. These are shown in Table and are defined in RFC 793.

Note that multiple codes, commonly called flags, can be sent in the same segment header, such as SYN and ACK (SYN/ACK), or FIN and ACK (FIN/ACK).
## TCP Control Information

<table>
<thead>
<tr>
<th>TCP Message</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ack</td>
<td>Acknowledges receipt of data</td>
</tr>
<tr>
<td>Fin</td>
<td>Terminates a connection</td>
</tr>
<tr>
<td>Psh</td>
<td>Acts as the push function</td>
</tr>
<tr>
<td>Rst</td>
<td>Resets the connection</td>
</tr>
<tr>
<td>Syn</td>
<td>Initiates a connection and synchronizes sequence numbers</td>
</tr>
<tr>
<td>Urg</td>
<td>Points to urgent data in the segment payload</td>
</tr>
</tbody>
</table>
In this situation, the packet-filtering firewall examines not only the source and destination addresses and port numbers, but, for TCP connections, it also examines the code bits to determine whether this is traffic being initiated from a device or traffic being sent in response to a request.
For example

- When the internal user (200.1.1.10) sends a TCP SYN, you know that the 170.1.1.1 will respond with a SYN and ACK in the TCP segment header.

- Therefore, if you know what kind of response control flags TCP uses, you could configure your packet-filtering firewall to allow this traffic, as shown in Figure.
Packet-Filtering Firewall Example—Examining Transport Control Codes
Two problems exist with examining control codes at the transport layer:

- Not all transport layer protocols support control codes.
- Control codes can be manipulated manually to allow a hacker to slip packets through a packet-filtering firewall.
(cont.)

- One of the biggest problems of having the packet-filtering firewall examine the control codes is that, in the TCP/IP protocol suite, TCP has control codes, but UDP doesn't.
However, the packet-filtering firewall cannot distinguish between a valid response and a fake response.

With a fake response, a hacker generates TCP segments with certain code flags set, trying to gain access through your firewall.

- A packet-filtering firewall, cannot distinguish between the two types of traffic.
State Table

- Unlike packet-filtering firewalls, stateful firewalls use a mechanism to keep track of the state of a connection.

- See Figure and Figure for an illustration of this.
Stateful Firewall Filtering Example—Part 1
Stateful Firewall Filtering Example—Part 2
In this example, the packet-filtering firewall has been replaced by a stateful firewall, but the filtering rule is unchanged: Any traffic sent to 200.1.1.10 is dropped.
(cont.)

- Assume that 170.1.1.1 sends traffic to 200.1.1.10. As shown in Figure, this traffic is dropped.

- Now assume that 200.1.1.10 opens a web connection to 170.1.1.1, as shown in the bottom part of Figure.
  - When 200.1.1.10 does this, it uses a TCP segment with a source port of 10,000 and a destination port of 80. It uses a SYN flag in the control field.
(cont.)

- When the stateful firewall receives this traffic, it first checks to see whether the 200.1.1.10 connection is allowed out.

- In this case, no filtering rules prevent this. Unlike a packet-filtering firewall, which just forwards the packet to 170.1.1.1, a stateful firewall adds a filtering rule to its configuration.
This information either is added to the top of the existing filtering rule set or is placed into a state table.

This table is used to keep track of the state of connections.

The former process is shown in Figure.
(cont.)

- After 170.1.1.1 receives the connection request, it responds to 200.1.1.1 with a SYN/ACK.

- When this segment reaches the stateful firewall, the firewall looks in its state table first (if the second method discussed previously is used) to see if the connection exists.
Then it processes the filtering rules on the interface.

In this example, only one table was used, but the connection entry was placed at the top. Because the connection information was added when 200.1.1.1 initiated the connection, the stateful firewall knows that the response from 170.1.1.1 (TCP port 80) to 200.1.1.1 (TCP port 10,000) is part of an existing connection and, therefore, that should allow the traffic, as shown in Figure.
One advantage of the stateful process is that when the connection terminates, the source or destination device tears down the connection and the stateful firewall notices this by examining the TCP header control flags and dynamically removes the connection from the state table (or filtering rules table).
Therefore, when comparing packet-filtering firewalls and stateful firewalls, stateful firewalls are more intelligent because they understand the state of a connection:

- initiating a connection,
- transferring data,
- or terminating a connection.

Basically, a stateful firewall contains a superset of packet-filtering functions.
Advantages of Stateful Firewalls

- Stateful firewalls are aware of the state of a connection.
- Stateful firewalls do not have to open up a large range of ports to allow communication.
- Stateful firewalls prevent more kinds of DoS attacks than packet-filtering firewalls and have more robust logging.
First

- Stateful firewalls typically build a state table and use this table to allow only returning traffic from connections currently listed in the state table.
  - After a connection is removed from the state table, no traffic from the external device of this connection is permitted.
  - Therefore, these types of connections are more difficult to spoof.
Stateful firewalls do not require you to open a large range of port numbers to allow returning traffic back into your network: The state table is used to determine whether this is returning traffic; otherwise, the filtering table is used to filter the traffic.
Third

- By using a state table, the stateful firewall can prevent more kinds of DoS attacks than a packet-filtering firewall.
- Plus, the stateful firewall can log more information than a packet-filtering firewall, such as when a connection was set up, how long it was up, and when it was turn down.
Limitations of Stateful Firewalls

- They can be complex to configure.
- They cannot prevent application-layer attacks.
- They do not support user authentication of connections.
- Not all protocols contain state information.
- Some applications open multiple connections, some of which use dynamic port numbers for the additional connections.
- Additional overhead is involved in maintaining a state table.
Stateful Firewall Problem: Nonstateful Protocols

- In addition to these problems, stateful firewalls have issues with nonstateful protocols.
  - Protocols that go through a defined process to establish, maintain, and tear down a connection are called stateful; mechanics are defined as to how these processes occur. TCP is an example of a stateful protocol.
However, not all protocols are stateful: UDP and ICMP are not.

For example, UDP has no defined process for how to set up, maintain, and tear down a connection; this is defined on an application-by-application basis.
In most of these applications, many packets are sent between the source and destination, typically at a constant rate. Most stateful firewall solutions treat UDP traffic as stateful by assigning an idle timer to these connections in the state table.
(cont.)

- As an example, a stateful firewall might use an idle timer of 30 seconds; if after 30 seconds no UDP traffic is seen for a UDP entry in the state table, the stateful firewall removes it.
The main problem with this approach is that if a hacker sends spoofed packets into your network, this would keep the entry in the table indefinitely.

Of course, a hacker must be quick about this because most UDP connections are temporary.
Stateful Firewall Problem: Multiple Application Connections

- Another problem that stateful firewalls have involves dealing with applications that open additional connections to transmit information.
  - These can include FTP, multimedia, NetBIOS, and many others.
  - FTP is used as an example here.
(cont.)

- FTP supports two different modes:
  - Standard (or active)
  - Passive
- Both modes set up two TCP connections. An example of these connections is shown in Figure.
FTP Connections

Standard FTP

TCP Port = 1024
TCP Port = 21

FTP Server

TCP Port = 20

Passive FTP

TCP Port = 1024
TCP Port = 21
TCP Port = 1024
TCP Port = 21

FTP Server

TCP Port = 20
With passive-mode FTP, as long as the user is inside the network establishing connections going out, you have no problems: Both outbound connections are placed in the state table, and the returning traffic for these automatically is allowed.

However, if the client device is outside the stateful firewall, you would need a specific filtering rule to allow the port 21 connection (called the control channel) and a very expansive filtering rule to allow the second connection (the data channel).
With standard FTP, if the client is inside the network and the server is outside, both stateful and packet-filtering firewalls would have problems dealing with the data connection that the FTP server was establishing to the client: You would have to open a whole range of ports to allow this second connection.
Stateful Firewall Problem: Size of State Table

- When it comes to the state table, it is a double-edged sword for stateful firewalls.
  - But in large networks, the stateful firewall might be busy building and maintaining the state table, putting an extra burden on its processing capacity.
  - The more connections your stateful firewall must monitor, the more horsepower your stateful firewall needs to maintain the table, thus increasing its cost.
Uses for Stateful Firewalls

Because of its increased intelligence over packet-filtering firewalls, stateful firewalls typically are used in the following areas:

- As a primary means of defense
- As an intelligent first line of defense (perimeter router with stateful capabilities)
- Where more stringent controls over security than packet filtering are needed, without adding too much cost
3.4. Application Gateway Firewalls

- Application gateway firewalls (AGFs), commonly called *proxy firewalls*, filter information at Layers 3, 4, 5, and 7 of the OSI reference model, as shown in Figure.

- Because AGFs process information at the application layer, most of the firewall control and filtering is done in software, which provides much more control over traffic than packet-filtering or stateful firewalls.
Application Gateway Firewalls and the OSI Reference Model
Sometimes AGFs support only a limited number of applications, or even just one application.

Some of the more common applications that an AGF might support include e-mail, web services, DNS, Telnet, FTP, Usenet news, LDAP, and finger.
Authentication Process

- One of the features of AGFs is that they typically allow you to authenticate connection requests before allowing the traffic to an internal or external resource.

- This enables you to authenticate the user requesting the connection instead of the device.
(cont.)

- This is one disadvantage that packet-filtering and stateful firewalls have: They examine only Layers 3 and 4 information and, thus, can authenticate only the Layer 3 address of a device.
(cont.)

- **Figure** shows a simple example of an AGF using an authentication process.
- In this example, the user first must authenticate to the AGF.
AGF Authentication Process
This can be done by having the user open a special connection—perhaps a web browser connection to the AGF, or the AGF can intercept the user's initial connection request and send the user a request for authentication information, like a web browser pop-up window.
The AGF or an authentication server then authenticates the user's identity.

- The authentication process occurs in software at the application layer.

In Figure, the authentication database is on the AGF and uses a username and password. In this database, the AGF allows Richard to access web server A upon successful authentication, but it will not allow Richard to access web server B.
NOTE

- To make the authentication and connection process more efficient, many AGFs authenticate a user once and then use authorization information stored in the authentication database to determine what resources a person can access.
Authentication Methods

- An AGF can use many methods to authenticate a connection request, including username and passwords, token card information, Layer 3 source addresses, and biometric information.
(cont.)

- Typically, Layer 3 source addresses are not used for authentication, unless they are combined with one of the other methods.
  - Authentication information can be stored locally or on a security server or directory service.
If you are using a username and password for authentication, the AGF prompts for the username and password.

One problem with this authentication method is that if the username and password are sent across the connection in clear text, this information is susceptible to eavesdropping.
(cont.)

- Therefore, this information should be encrypted. Typically, this is done through the Secure Socket Layer (SSL) protocol within a web browser connection.
Application Gateway Firewall Types

- AGFs fall under two categories:
  - Connection gateway firewalls (CGFs).
  - Cut-through proxy (CTP) firewalls.
Connection Gateway Firewalls

- CGFs offer more protection than CTP firewalls.

- **Figure** shows the process that a person goes through when setting up a connection through a CGF.
Connection Gateway Firewall Process

Step 1: Authentication Inbound

<table>
<thead>
<tr>
<th>User</th>
<th>Destination</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Richard</td>
<td>200.1.1.2</td>
<td>Permit</td>
</tr>
</tbody>
</table>

Step 2: Open a Connection to the Internal Web Server

Step 3: Examine Application Layer Traffic from Richard’s Connection and Pass it to the Web Connection

Web Server 200.1.1.2

Outside User Richard 170.1.1.1
NOTE

- Many CGFs (and CTPs) enable you to configure multiple authorization rules for a single user.
- Therefore, when the user successfully authenticates, all the authorization rules are put into effect without requiring the user to authenticate for each connection request.
One nice feature of a CGF is that it can examine all data that Richard sends to the web server, even specific URL requests. This allows the CGF to examine what pages Richard tries to access and whether Richard is trying to sneak malformed URLs or data that might try to crash the server or open the server because of a security weakness.
Cut-Through Proxy Firewalls

- One of the main problems of a CGF is that, for the applications that it supports, all traffic is processed at the application layer; this is very process-intensive.

- In some cases, you might be interested only in performing authentication of a connection at the application layer.
Of course, you could perform this function with a CGF; however, a CGF always processes information at Layer 7, which can introduce a noticeable delay in individuals' connections, especially on an CGF that handles thousands of connections.
Cut-through proxy (CTP) firewalls are a modified version of CGF that deals with this inefficiency.

Figure shows a simple example of the process that a CTP uses to allow connections into a network.
Cut-Through Proxy Firewall Process

**Step 1:** Authentication Inbound

<table>
<thead>
<tr>
<th>User</th>
<th>Destination</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Richard</td>
<td>200.1.1.2</td>
<td>Permit</td>
</tr>
</tbody>
</table>

**Step 2:** Add Filtering Rule

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>170.1.1.1</td>
<td>200.1.1.2</td>
<td>TCP 80</td>
<td>Permit</td>
</tr>
</tbody>
</table>

**Step 3:** Process Traffic at Layers 3 and 4

1. Workgroup Switch
2. Cut-through Proxy Firewall
3. Web Server 200.1.1.2
4. Outside User Richard 170.1.1.1
In this example

- Richard tries to access the internal web server (200.1.1.2).
  - The CTP intercepts the connection request and authenticates Richard, shown in Step 1.
  - After authentication, this connection and any other authorized connections are added to the filtering rules table, shown in Step 2.
  - From here, any traffic from Richard to the web server is handled by the filtering rules at Layers 3 and 4.
As you can see from this example, the authentication process is handled at Layer 7; after being authenticated, however, all traffic is processed at Layers 3 and 4.

Therefore, the advantage that CTP has over CGF is a huge boost in throughput.

However, because CTP does not examine application-layer data, it cannot detect application-layer attacks.
Typically, the CTP supports Telnet, HTTP, and HTTPS for handling the initial authentication.
Advantages of Application Gateway Firewalls

- They authenticate individuals, not devices.
- Hackers have a harder time with spoofing and implementing DoS attacks.
- They can monitor and filter application data.
- They can provide detailed logging.
Limitations of Application Gateway Firewalls

- They process packets in software.
- They support a small number of applications.
- They sometimes require special client software.
The main limitation of AGFs is that they are very process intensive.

To address these issues, you can use one of these two solutions:

- Use a CTP
- Have the AGF monitor only key applications
Other Types of Application Proxy Devices

- Other types of application gateway devices exist besides AGFs.
- AGFs are used mainly for security purposes; however, other application gateways (commonly called *proxies*) can be used to help with throughput issues.
For example, a common type of proxy is an HTTP proxy. With an HTTP proxy, an individual configures the web browser to point to the proxy. Whenever the individual requests a web page, the request goes to the proxy first.
Sometimes these proxies are used to help reduce logging functions on the AGF itself.

This is important if you have acceptable use and abuse policies and need to monitor resource requests so that you can enforce these policies.
Uses for Application Gateway Firewalls

- A CGF commonly is used as a primary filtering function.
- A CTP commonly is used as a perimeter defense.
- An application proxy is used to reduce the logging overhead on the CGF, as well as to monitor and log other types of traffic.
3.5. Address-Translation Firewalls

- Address translation was developed to address two issues with IP addressing:
  - It expands the number of IP addresses at your disposal.
  - It hides network addressing designs.
The main reason that address translation (RFC 1631) and private addresses (RFC 1918) were developed was to deal with the concern of the shortage of addresses that was seen on the horizon in the mid-to late 1990s.
Basically, address translation translates the source/destination address(es) and/or port numbers in an IP packet or TCP/UDP segment header.

Because of this, address-translation firewalls (ATF) function at Layers 3 and 4 of the OSI reference model, as shown in Figure.
Address-Translation Firewalls and the OSI Reference Model
Filtering Process

- Most people assume that address translation is used to translate private to public addresses or vice versa, so you might be wondering how you can use address translation as a security function.
(cont.)

- Examine Figure, which illustrates the usefulness of address translation in protecting your network.
- In this example, two web servers have private addresses assigned to their NICs, 192.168.11.2 and 192.168.12.2.
Address-Translation Firewall Example

![Diagram showing address translation rules.](image)

<table>
<thead>
<tr>
<th>Internal Private Address</th>
<th>Internal Public Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.11.2</td>
<td>200.1.1.2</td>
</tr>
<tr>
<td>192.168.12.2</td>
<td>200.1.1.3</td>
</tr>
</tbody>
</table>
Because private IP addresses are nonroutable in public networks, a public address must be associated with these two devices, and a DNS server needs to send the public address in response to DNS queries for the addresses of these devices.
(cont.)

- The ATF defines the translation rules.
- Traffic heading to 200.1.1.2 is translated to 192.168.11.2, and traffic to 200.1.1.3 is translated to 192.168.12.2, and vice versa.
This process serves two functions

- First, an outside person cannot decipher anything about the IP address structure of your network:
  - That person knows only that 200.1.1.2 and 200.1.1.3 are reachable addresses and appear to be on the same segment.
  - The outside person does not know that these web servers are on two different physical segments behind two different routers.
(cont.)

- Second, traffic sent to any other device in your network cannot be reached unless it first is translated; remember that your internal devices are using private addresses.
Advantages of Address-Translation Firewalls

- They hide your network-addressing design.
- They control traffic entering and leaving your network.
- They allow for the use of private addressing.
Limitations of Address-Translation Firewalls

- Delay is introduced because of packet manipulations.
- Some applications do not work with address translation.
- Tracing and troubleshooting become more difficult.
Uses for Address-Translation Firewalls

- When you have a private IP addressing scheme in your internal network
- When you need to easily separate two or more networks
3.6. Host-Based Firewalls
Advantages of Host-Based Firewalls

- They can be used to enhance your security.
- Some can provide host-based authentication.
- Their cost is typically less than $100—and sometimes they even are free.
Limitations of Host-Based Firewalls

- They are software-based firewalls.
- They are simplified packet filters.
- They have weak logging capabilities.
- They are difficult to manage on a large scale.
Uses for Host-Based Firewalls

- With home users or telecommuters with Internet access
- In small SOHO environments
- To add an extra level protection to critical resources, such as e-mail and database servers
3.7. Hybrid Firewalls

- Because of the many advances in technology, the widespread use of the Internet, and the explosion of e-commerce and e-business, the need for security has increased greatly.
- Therefore, classifying a firewall product is a difficult, if not impossible, process.
4. Firewall Design

- You should follow five basic guidelines when designing a firewall system:
  - Develop a security policy.
  - Create a simple design solution.
  - Use devices as they were intended.
  - Implement a layered defense to provide extra protection.
  - Consider solutions to internal threats that should be included in your design.
Developing a Security Policy

- One of the first things you do when designing a firewall system is to create a security policy.
- The policy should define acceptable and unacceptable behavior, should state restrictions to resources, and should adhere to the company's business plan and policies.
The key to a good design is basing it on a security policy.

Basically, a policy defines who is allowed to access resources, what they are allowed to do with resources, how resources should be protected (in general terms), and what actions are taken when a security issue occurs.
The resources that require access from internal and external users
The vulnerabilities associated with these resources
The methods and solutions that can be used to protect these resources
A cost-benefit analysis that compares the different methods and solutions
Designing Simple Solutions

- A firewall system design should be kept simple and should follow your security policy.
  - The simpler the design is, the easier it will be to implement it, maintain it, test and troubleshoot it, and adapt it to new changes.
Using Devices Correctly

- Network devices have functional purposes; they were built with a specific purpose in mind.

- Using the wrong product to solve a security problem can open you to all kinds of security threats.
Creating a Layered Defense

- A security design typically uses a layered defense approach.

- In other words, you usually do not want one layer of defense to protect network.
  - If this one layer is compromised, your entire network will be exposed.
A Medieval Firewall System
Dealing with Internal Threats

- Too often, security personnel are concerned about protecting a company's resources and assets from outside threats.
  - Remember that it is much easier to attack your assets from within; plus, most threats and attacks (60 to 70 percent) are internal attacks.
DMZ

- Most firewall systems use a demilitarized zone (DMZ) to protect resources and assets.
- A DMZ is a segment or segments that have a higher security level than that of external segments, but a lower security level than that of internal segments.
DMZs are used to grant external users access to public and e-commerce resources such as web, DNS, and e-mail servers without exposing your internal network.

A firewall is used to provide the security-level segmentation among the external, DMZ, and internal resources.
Security Level Example

Diagram showing the relationship between user PCs, database servers, remote company servers, firewalls, and network servers within an internal network and DMZ (Demilitarized Zone) segment connected to the internet.
The firewall has the following four interfaces:

- A connection to the Internet, assigned a low security level
- A connection to the DMZ, where public servers are located, assigned a medium security level
- A connection to a remote company that is working on a project for them, assigned a low security level
- A connection to the internal network, assigned a high security level
This company has assigned the following rules:

- High- to low-level access: permit
- Low- to high-level access: deny
- Same-level access: deny
Given these rules, the following traffic is allowed automatically to travel through the firewall:

- Internal devices to the DMZ, the remote company, and the Internet
- DMZ devices to the remote company and the Internet
DMZ Types

- You can have a single DMZ, multiple DMZs, DMZs that separate the public network from your internal network, and DMZs that separate traffic between internal networks.
Single DMZ

- Single DMZs come in two types:
  - Single segment
  - Service-leg segment
Single DMZ with a Single Segment
Single DMZ with a Service-Leg Segment
Two advantages over the single-segment DMZ

- The firewall sometimes can be connected directly to the Internet, removing the extra cost of the perimeter router.
- All security-level polices can be defined on one device (in a single-segment DMZ, you must define your policies on two devices).
Multiple DMZs

- Firewall system can be used to separate multiple areas of your network, including multiple DMZs
Multiple DMZ Example
Internal DMZ

- Another type of DMZ is an internal one.
- An internal DMZ enables you to provide separation between different parts of your internal network.
Internal DMZ Example

[Diagram of a network with labeled nodes: Engineering Server, Engineering, Medium, Low, Firewall, Accounting, Accounting Server, Other Internal Users, Internal Network]
Components

- A good firewall system typically contains the following components:
  - Perimeter router
  - Firewall
  - VPN
  - IDS
Firewall Component

- The functions of the firewall can include the following:
  - Stateful filtering
  - User authentication of connection with CTPs
  - Connection filtering with CGFs
  - Address translation
Simple Firewall System Design
Enhanced Firewall System Design