Network Security

Professor Gaurav S. Kasbekar

Department of Electrical Engineering

Indian Institute of Technology, Bombay

Week - 09

Lecture - 54

Firewalls and Intrusion Detection Systems: Part 4

Hello, recall that in the previous lecture, we discussed traditional packet filters. We now discuss stateful packet filters and application gateways. Recall the example access control list that we discussed in the previous lecture. There is an organization whose hosts have IP addresses of this form. The prefix is 222.22.

- Consider an organization whose hosts have IP addresses of the form 222.22/16
- The first two rules together allow internal users to surf the Web
  - but external sources are not allowed to establish a TCP connection with a Web server inside the organization
- The third and fourth rules together allow DNS packets to enter and leave the organization's network

Action	Source address	Dest. address	Protocol	Source port	Dest. port	Flag bit
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80	any
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	--
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	--
deny	all	all	all	all	all	all

We saw in the previous lecture that the first two rules, that is, these two, allow internal users to surf the web, but external sources are not allowed to establish a TCP connection with the web server inside the organization. And these third and fourth rules together allow DNS packets to enter and leave the organization's network. Now, recall that in a traditional packet filter, filtering decisions are made on each packet in isolation. In contrast, stateful packet filters track TCP connections; that is, they know which TCP connections are active between a host inside the organization's network and a host outside the network, and they use this knowledge to make filtering decisions. So, stateful packet filters maintain a table in which all the TCP connections that are active, they are stored.

Recall the access control list on the previous slide. It allows any packet arriving from outside the organization's network with the ACK bit set to one and source port 80 to pass

through the filter. Let's go back to the access control list. We can see that this second rule allows packets arriving from outside the network with source port 80 and the ACK bit equal to 1. These are allowed to pass through the firewall.

So, this fact can be used by an attacker. The attacker can send malformed packets to internal systems which may crash when they see unexpected header values. There is no active web session between a host inside the network and a particular web server. But an attacker may send a packet from that with the IP address of that web server to the internal host. And that packet will have source port 80 and ACK bit set to 1.

So, even though there is no ongoing web connection to a particular web server outside the network, an attacker can send a packet as if it is from the web server and it is part of an active web connection. So, this is a malformed packet. It is just meant to crash the internal host. So, an attacker can send malformed packets to internal systems and the internal systems may crash when they see unexpected header values. This can happen because the traditional packet filter does not know that there is no ongoing TCP connection between a host inside the network and external web server.

Traditional packet filters do not have state information about which TCP connections are active. So, one solution is modify the access control list so that it now blocks incoming packets with ACK bit set to 1 and source port 80. But the shortcoming is that this will prevent the organization's internal users from surfing the web. So, even packets which belong to legitimate connections to web servers outside the network, these packets will also be dropped. This motivates the need for stateful packet filters.

- Possible because firewall can observe:
  - ❑ the beginning of a new connection by observing a three-way handshake (SYN, SYNACK and ACK)
  - ❑ the end of a connection when it sees a FIN packet for the connection
- An example connection table of a firewall is shown in fig. below
  - ❑ indicates that there are three ongoing TCP connections, all Web connections initiated from within the organization

Source Address	Destination Address	Source Port	Destination Port
222.22.1.7	37.96.87.123	12699	80
222.22.93.2	199.1.205.23	37654	80
222.22.65.143	203.77.240.43	48712	80

Stateful packet filters solve this problem by tracking all ongoing TCP connections in a connection table. That is, they store the source IP address, destination IP address, and other

details of all active TCP connections in a table. And then the stateful packet filter can consult this table to find out whether a particular packet is part of an active TCP connection or not. So, stateful packet filters can track all ongoing TCP connections. This is possible because the firewall can observe the beginning of a new connection by monitoring a three-way handshake consisting of SYN, SYN-ACK, and ACK packets.

We discussed the three-way handshake earlier. So, every new connection always begins with a three-way handshake and by observing these packets which are part of the three-way handshake, the firewall can observe the beginning of a new connection. The firewall can also detect the end of a connection when it sees a FIN packet for that connection. So, when the connection ends, the firewall can then remove the connection from the list of active connections. So, whenever a new connection is established via three-way handshake, the firewall adds the details of that connection to the table and whenever the connection ends via the exchange of FIN packets, the firewall removes the connection from the table.

Here is an example connection table of a firewall. So, one of the TCP connections has this source IP address, which is within the organization's network, and this destination address, which is an IP address outside the network. The source port is this one, which is a port number greater than 1023. And the destination port is 80. That means it corresponds to a web traffic session.

So, this is a TCP connection from a host inside the network to a web server outside the network. Similarly, these other three rows are also, they correspond to TCP connections from hosts inside the organization's network to web servers outside the organization's network. So, this connection table indicates that there are three ongoing TCP connections, which are all web connections initiated from within the organization. Now, the access control list is extended by adding a new column to it. That column is "check connection."

The check connection column is shown in this figure. This is added to the access control list. This indicates that the connection should be checked for two of the rules. So, if a packet matches either this rule or it matches this rule then the connection should be checked. That is, the firewall should consult the connection table, which stores the list of active TCP connections, and check whether this packet belongs to that active TCP connection or not.

So, the "check connection" field is marked only for packets which are sent from a host outside the network to a host inside the network. So, this corresponds to responses from the web server to the host inside the network, and this corresponds to responses from the DNS server to the host inside the network. So, these should be checked. We now study

some examples to see how the connection table and the extended access control list can be used together. Here is the first example.

Suppose an attacker attempts to send a malformed packet into the organization's network by sending a packet with TCP source port 80 and the ACK bit set. So, this packet looks as if it is being sent by a web server outside the organization's network in response to some request for web service by a host inside the network. And suppose the destination port is 12543 and the source IP address is 150.23.23.155, which is an IP address outside the organization's network. When this packet reaches the firewall, the firewall checks the access control list, which indicates that the connection table must be checked. Let's go back to that access control list.

This packet matches the second row of the access control list, and that indicates that the connection should be checked. Then the firewall checks the connection table, which is given here, and the firewall sees that this packet is not part of an ongoing TCP connection and hence drops the packet. So, we can see from here that there's no active TCP connection with this source IP address, that is, 150.23.23.155. So, this does not match any of the active TCP connections in this list. Also, this port number 12543 does not match any of the source ports here.

Hence, the firewall finds out that this packet is not part of an ongoing TCP connection and hence drops the packet. Here's another example. Suppose an internal user wants to surf an external website. First, the user's host sends a TCP SYN packet to the web server. This TCP connection gets recorded in the connection table.

There is an active TCP connection from the internal user's system to the external website. So that TCP connection gets recorded in the connection table. When the web server subsequently sends packets to the user's host with the ACK bit set, again those packets match the second row in that access control list. So, the firewall checks the table and sees that the corresponding connection is in progress. Hence, the firewall lets these packets pass and hence the firewall does not interfere with the internal user's surfing activity.

In this way, the packets are transmitted, they are not blocked by the firewall and hence this achieves our objective of letting internal users surf an external website successfully. This illustrates again how by checking the connection table, a firewall can find out whether the packets are part of a legitimate TCP connection or they are some malformed packets being sent just to crash some host inside the organization's network. So, we discussed how TCP connection can be handled, but DNS traffic is sent over UDP. So, how do we find out

whether a particular packet is part of a legitimate DNS exchange or it is some malformed packet? So, we need to also handle UDP traffic.

- E.g.: recall the rows corresponding to DNS traffic in extended access control list discussed earlier (see fig.)

Action	Source Address	Destination Address	Protocol	Source Port	Destination Port	Flag Bit	Check Connection
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80	any	
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK	X
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	-	
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	-	X
deny	all	all	all	all	all	all	

Recall that it is easy to maintain a connection table for TCP connections because the connection starts when a three-way handshake takes place and it ends when FIN packets are sent. But UDP traffic is not connection-based. There is no three-way handshake before the UDP traffic is sent and there are no FIN packets sent after the UDP traffic is exchanged. So, this approach cannot be used in the case of UDP. To handle UDP traffic, a stateful packet filter tracks the state using only the source and destination IP addresses and the source and destination port numbers.

So, for example, if a host inside the organization's network sends a query to a DNS server outside the organization's network, then the stateful packet filter can store the state, which means that the state says that a host from inside the organization's network has just sent a DNS query to a DNS server from outside the organization's network. So, later on, when the DNS server sends a response to the user inside the organization's network, the stateful packet filter will come to know that this is part of a legitimate DNS exchange that is happening. As an example, recall the rows corresponding to DNS traffic in the extended access control list discussed earlier. These rows are the third and fourth rows. So, these correspond to DNS traffic.

Now, for this row, the table says "check connection." And that can be checked as we just mentioned. When a host from inside the organization's network sends a DNS query, the stateful packet filter will store the source IP address, destination IP address, source port number, and destination port number and it will know that a DNS query has been sent. So, when a packet is sent from the DNS server to the host inside the organization's network, then the stateful packet filter will check the connection and it will find out that this is part of a legitimate DNS packet exchange. So, it will allow the packets to pass.

In this way, we can ensure that a stateful packet filter can track the state even for UDP packets. So, a stateful packet filter can handle TCP packets as well as UDP packets. This concludes our discussion of stateful packet filters. Next, we discuss the third type of firewall: the application gateway. To motivate the need for application gateways, recall that traditional packet filters and stateful packet filters perform filtering on the basis of the contents of IP, TCP/UDP, and ICMP headers.

But traditional packet filters and stateful packet filters, they do not look at application data. But suppose an organization wants to implement the following policies. It wants to allow Telnet to external hosts to a restricted set of internal users as opposed to IP addresses. For example, there may be different users, such as Alice and Bob; they may be connecting from different IP addresses. So, the organization wants to allow certain users from a restricted set of internal users; it wants to allow them to Telnet to external hosts.

So, the decision of whether to allow a particular Telnet connection is based on which user has made that request and it is not based on which IP address has made that request. The organization wants such privileged users to authenticate themselves first before being allowed to create Telnet sessions to external hosts. So, this authentication also needs to be performed and cannot be performed by the traditional packet filters or stateful packet filters. So, we need another kind of firewall to achieve this. So, how can these policies be implemented using packet filters?

They cannot be implemented using packet filters because the information about the identities of the users is application layer data and it is not included in any of these headers, IP, TCP/UDP or ICMP. So, all these headers are examined by traditional packet filters and stateful packet filters. But information about identities of users is application layer data which is not examined by traditional packet filters and stateful packet filters. To implement the above policies, application gateways are required. Application gateways make policy decisions based on application data as opposed to headers, such as IP, TCP/UDP and ICMP.

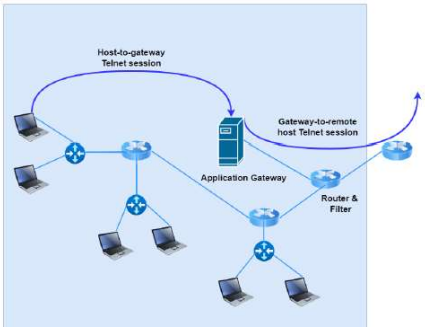
Let’s discuss how application gateways function. An application gateway is an application specific server. There is a different application gateway for web traffic, there is a different application gateway for FTP traffic, and so on and so forth. All application data (inbound and outbound), must pass through the application gateway. For example, if there is an application gateway for Telnet, in that case all Telnet application data (inbound as well as outbound) must pass through the application gateway.

It cannot bypass the application gateway and pass through the firewall. Then the firewall will drop the packets. Multiple application gateways corresponding to different applications can run on the same host. For example, there may be one application gateway corresponding to Telnet, one corresponding to FTP, and one corresponding to web traffic. So, such multiple application gateways can run on the same host.

But each gateway is a separate server with its own process. Here is an example which illustrates the operation of application gateways. Suppose we want to design a firewall, which allows only a restricted set of internal users to Telnet to external hosts. It prevents all external clients from Telneting to an internal host. So, this policy can be implemented as shown in this figure, using a combination of a packet filter in a router and a telnet application gateway.

This is the organization's network. All the nodes that are within this blue box, they are in the organization's network and this particular node is outside the organization's network. This is the application gateway and this is a packet filter in a router. Consider a host from inside the organization's network which wants to establish a Telnet connection to a host outside the organization's network. Then that Telnet connection has to pass through this application gateway as shown in this figure.

- This policy can be implemented using (see fig.) a combination of:
  - ❑ a packet filter (in a router) and
  - ❑ a Telnet application gateway
- The router's filter is configured to block all Telnet connections except those that originate from the IP address of the application gateway
  - ❑ hence, inbound Telnet connections are blocked



Some host from outside the organization's network cannot set up a Telnet connection to a host inside the organization's network. So, this can be achieved as follows. The router's filter is configured to block all Telnet connections except those that originate from the IP address of application gateway. So, only if there is an Telnet connection request from this host, in that case, the packets will be passed by this router. If some other host such as, if this host, for example, directly tries to send a Telnet packet to a host outside the organization's network, then those packets will be dropped by the router.

So, the Telnet request has to be initiated by the application gateway only. Hence, inbound Telnet connections are automatically blocked. Because if some host from outside the organization's network is sending Telnet request packets, then they will be dropped by this packet filter. Since those packets are not coming from the application gateway, which is inside the organization's network. Now, consider an internal user who wants to Telnet to an external host.

So, that internal host is shown here. The user must first set up a Telnet connection with the application gateway. This shows a Telnet connection with the application gateway. That is shown by this curve. The application gateway prompts the user for the username and password.

Recall that only users from a restricted list of users should be allowed to host outside the organization's network. So, the application gateway prompts the user for the username and password. When this information is supplied, the gateway checks to see if the user has permission to Telnet to an external host. So, once the user provides the username and password, the identity of the user can be checked by the gateway. And the gateway can verify whether the user has permission to Telnet to an external host.

If the user does not have permission to Telnet to an external host, then the Telnet connection from the internal user to the application gateway is terminated by the gateway. In this way, users who are not from the set of restricted users, their Telnet connections are blocked by this combination of application gateway and packet filter. If the user has permission, the gateway prompts the user for the host name of the external host they want to connect to. So, that external host is somewhere outside the network and the gateway prompts the user for the host name of the external host to which the user wants to connect. And then it sets up a Telnet connection between the gateway and the external host.

So, the gateway establishes a Telnet connection from itself to an external host to which this host wanted to establish a Telnet connection. So, this Telnet connection is allowed by the packet filter because it is initiated by the application gateway. Subsequently, the application gateway relays packets between the external host and the user. So, if the external user to which the Telnet connection is established is somewhere here and the other end of the Telnet connection is this one, then packets are exchanged between this user and the external user and these packets are relayed by the application gateway. So, the application gateway accesses the relay and forwards packets from this user to the user outside the network and vice versa.

So, notice that in step 2, The packet filter permits this step 2 because the gateway initiates the Telnet connection to the external host. If this host had tried to directly establish a Telnet connection to the external host, then those packets would have been dropped by the router. So, the packet filter permits this step 2 because the gateway initiates the Telnet connection to the external host. In this way, we can enforce the requirement that only users from a restricted list of users are allowed to establish Telnet connections to hosts outside the organization's network. And inbound Telnet connections have to be blocked.

So, these requirements can be enforced in this way using an application gateway. Let's discuss some more about application gateways. Networks of organizations often have multiple application gateways. For example, they may have gateways for different application gateways for Telnet, HTTP, and FTP. So, an application gateway is specific to a particular application.

HTTP gateways are very common in organization networks. Apart from the basic function of monitoring the HTTP traffic and allowing HTTP traffic only to users from a restricted list, the HTTP gateway also performs some other functions. It includes a web cache. That is, recently visited web pages are cached so that they do not have to be repeatedly fetched from external websites. So, for example, many users from inside the organization's network may visit the website of Times of India, say.

So, when a user connects to Times of India's website, then the HTTP gateway downloads the webpage of Times of India, and then immediately subsequently, if another user attempts to connect to Times of India's webpage, then the HTTP gateway retrieves the webpage from its cache. It does not connect again to the website and fetch the data again. So, the use of a web cache can help in preventing the repeated fetching of the same data from external websites. So, HTTP gateways include a web cache which helps in efficiency. The HTTP gateway also scans incoming web pages for virus signatures and objectionable content.

So, if there's some malware which some external website is trying to send to an internal user, then that will be, those packets will be blocked by the HTTP gateway. And if some users attempt to download objectionable content, then that is also blocked by the HTTP gateway. This is possible because the HTTP gateway examines the application data, not just the headers in the packet. But there are some limitations of application gateways. One limitation is that we require a different application gateway for different applications.

A performance penalty needs to be paid because all data is relayed via the gateway. So, if a large number of users use a particular application, then the application gateway will be congested because a lot of traffic flows through it. This becomes a concern when a large number of users or applications use the same gateway machine. So, these are some limitations of application gateways. Later on, we'll discuss another kind of device called an intrusion detection system, which also examines application data.

We'll see later how intrusion detection systems overcome these limitations of application gateways. So, in summary, there are three types of firewalls, and in this lecture, we discussed stateful packet filters and application gateways. So, this concludes our discussion of the three types of firewalls, namely traditional packet filters, stateful packet filters, and application gateways. In the next lecture, we'll discuss intrusion detection systems. Thank you.