Chapter 9

Practical assignment 9

9.1 Background

Firewalls are usually deemed to operate on layer 3 or layer 7.

9.1.1 Layer 3 firewalls

A layer 3 firewall inspects network layer packets that flow through it. The firewall is typically installed on the host(s) through which a corporate or similar network is connected to the outside world. It is therefore in a position to monitor traffic and block certain traffic from flowing into the corporate network or leaving the corporate network. (In what follows, we will talk about a *corporate* network, though the principles apply in exactly the same manner to other networks, such school, home, and NGO networks. In fact, the principles also apply to subnetworks within such an entity, such as a department or even a single computer in the bigger organisation, if a firewall is installed for the subunit.)

From the layer 3 headers it can determine the source and destination addresses of the packets. The rules associated with the firewall may determine that the communication may be allowed or should be blocked. If it is to be allowed, the packet is forwarded to the next hop on its route. If the message is to be blocked, the packet may simply be dropped. As a simple example, if it is known that crackers operate from some known addresses, any traffic to and from those addresses can be blocked. Note that this example is unrealistic.

To properly use a layer 3 firewall the knowledge that the (immediate) payload of a layer 3 packet is a layer 4 packet helps. The first information that follows the layer 3 header in the layer 3 packet is, in principle, the header of the layer 4 packet it contains. Below we will consider why this is not always true, but until then we will proceed under the assumption that it is indeed always true.

The layer 3 firewall is therefore not only able to consider the layer 3 header, but also able to inspect the layer 4 header. This means it can, amongst others, also determine the source and destination ports of the packets. If the layer 4 content contains a TCP header, the firewall can also determine from the SYN, FIN and ACK flags determine the connection status of the TCP stream.

Now it becomes possible to restrict, say, web access to only be directed at the corporate web server: if the destination address is one that belongs to the corporate network and the destination port is 80, only allow the packet to proceed if the destination address is that of the corporate web server. In a similar manner, connections to other services may be restricted to the appropriate servers. At the very least the sysadmin can now ensure that the servers are properly configured and patched against the latest vulnerabilities for the services they provide.

A challenge remains: Someone in the corporation may operate, say, an unauthorised web server on some non-standard port on the corporate network. But the firewall is also of use in this case: Add a rule that blocks any connection to be opened from outside the organisation to a computer on the corporate network by blocking the TCP handshake (that is, any message for which the SYN flag is set, the ACK flag is reset, and the destination address is somewhere on the corporate network). It should be obvious that this will only be useful if this rule is only processed *after* rules allowing traffic to proceed to the appropriate servers have been processed. Stated differently, if a web request arrives and its destination is the corporate web server, ACCEPT it. Do the same for all other legitimate service requests. Then, as a default, REJECT all new requests to open a connection to any other hosts.

This still leaves us with some questions about those cases where the payload of a layer 3 packet is a UDP packet (or anything else, for that matter). We will not discuss those in this assignment.

We did promise to talk about those cases where the TCP header does not immediately follow the layer 3 header in the packet. This may happen where, say, a long TCP stream is transmitted, and it is split into several IP datagrams. The TCP header will follow the IP header in the first IP packet, but not in the subsequent IP packets. Fortunately, this problem is easy to solve — if one blocks the TCP header, the TCP stream can never connect, and the remaining part of the TCP stream is useless. Hence, blocking the TCP header is sufficient. When one explores this a bit deeper, there are all sorts of interesting games crackers play, and security specialists needs to be aware of, but we will not explore these in the current assignment.

9.1.2 Layer 7 firewalls

A layer 3 firewall provides one with many options to protect the corporate network from the big bad Internet out there, and even from corporate insiders to access services on the outside that are not deemed to be in the corporate interest. However, layer 3 is too far removed from the application layer to provide complete protection. By inspecting the ports, the layer 3 firewall knows which service (or application) is to be accessed, but further details are hidden in too many layers of payload — one cannot use the same trick we used to let a layer 3 firewall also access layer 4 headers to reach all the way to layer 7 details. Hence, layer 7 firewalls are typically deployed in addition to layer 3 firewalls. Layer 7 firewalls are often referred to by the term *application proxies*. Note that, despite the implication earlier in this paragraph that complete security may be achieved by increasing the 'reach' of firewalls, firewalls always offer only a small — albeit important — part of network security. One should carefully think about threats that are not addressed by *any* firewall and have a broad view of network security. This assignment only considers firewalls, though.

One of the best-known examples of a layer 7 firewall is a web proxy. Browsers make provision for one to enter details about the proxy. If a web proxy is used, browsing requests will be sent to the web proxy, rather than the destination host. The web proxy will then forward the request to the destination host and receive the response. It will then relay the response to the browser that sent the request initially. The web proxy is able to inspect the request sent by the browser and block it, if the corporate policy does not allow access to the specific website. One typically has to log into the web proxy, so unauthorised parties can be prevented from using the corporate network to surf the web. In addition, if users are only allowed a certain amount of web traffic per month, a tally of the traffic relayed can be kept and requests from that user can be blocked once the threshold is reached. As another example, it becomes simple to allow employees to access social networks over the lunch break, but block it at other times, when employees are supposed to be working.

Note that such an application proxy needs to be used in conjunction with a layer 3 firewall: The layer 3 firewall needs to block any outgoing traffic to port 80 (or other relevant ports), except when the traffic comes from the web proxy.

This introduction quickly leads to opinions about 'corporate censorship'; we will, however, not succumb to that temptation in this assignment. For this assignment, we accept that the corporate network belongs to the corporation, and they are free to formulate policies regarding the network and enforce them.

Note again that web browsers make provision for web proxies and the user is often unaware of the fact that a web proxy is in use. The web browser would normally send its request to the web server as specified by the URL entered into

the browser. However, when a proxy is in use, that request will be sent to the proxy. The proxy will perform whatever inspection it is configured to perform, and then forward the request to the server — or send an error message to the web browser. The response from the server is similarly relayed to the browser by the proxy. For other applications, the ability to redirect messages to a proxy are typically not build into the client. If one wants to use a proxy, some non-standard process to relay the messages via an application proxy may be required.

9.2 Your assignment

You are expected to build an application proxy. Note that the application proxy will act as both a server and a client: The usual client will connect to the proxy, as if the proxy were the server. This requires server functionality: It will have to wait on some specified port (such as 55555) to accept requests from 'normal' clients for that application. The application proxy will then perform whatever checks it needs to perform on the request. After completing the checks, it has to connect to the intended server, and act as a client to that server.

For this assignment you are expected to write a POP3 proxy. The logic (or that of your boss) is the following. The corporate customer email service is hosted somewhere in the cloud. Employees are allowed to read customer service emails only when they are in the office. Hence it has been decided that the password of the corporate service will not be given to any employees. They would rather be given usernames and passwords that are checked by a proxy server that is only accessible from the network in the office. This proxy server 'knows' the real credentials for the cloud email service. Once the user has been authenticated via the initial part of the POP3 conversation by the proxy server, the proxy server initiates a POP3 conversation with the cloud server to be authenticated using the real credentials. Once authenticated, the proxy server merely relays messages between the user and the cloud server back and forth, until the POP3 session is terminated.

Obviously, this 'solution' raises a number of questions about its effectiveness. Would it have any benefit if employees use laptops that they can take home? Perhaps only installed desktop computers are allowed to be used in the office... But users are then able to copy emails to memory stick that they can take home... What can be done to make this scheme effective? Does the idea have any merit at all? Please ponder these questions. However, as the organisation's programmer your main task is to implement the software as specified by management. So, start designing the program despite any concerns you may have.

To simplify matters the 'corporate POP3 server' does not have to be in the cloud. It may be any server that you can access via POP3. One option is to use a

POP3 server installed on the server computer you used for other assignments.

The specification suggested that multiple users, each with an own password, may be able to access the proxy server. For your assignment, a single used id with password (that both differ from the real access credentials) will suffice.

Ideally the proxy server should execute on its own server computer. However, use either your server computer or your client computer to host it.

You should use any mail client (such as Thunderbird) to read received email. This presents you with the challenge: To which SMTP server do you connect. A nice solution would be to also write an SMTP proxy, but that is not required for this assignment. You may therefore provide the details of the 'real customer service' SMTP server to your email client. *However, you have to provide the (different) POP3 proxy credentials to your client.*

To demonstrate that your proxy works, send emails to the 'corporate' email account, and show that your proxy can access the emails via the proxy using POP3.

Ideally such a proxy would print event messages to a log file. To simplify demonstration, let the proxy run in its own window and print verbose 'log' messages to the screen. This should enable an observer to quickly notice what is happening at any given moment — such as when the vatious connections are established and terminated, when commands are relayed (or blocked), etc. However, do not print entire email messages to this 'log', because such text would obscure rather than highlight events as the happen.

9.3 Challenges

Here are some challenges to increase your mark.

1. If you implement an appropriate SMTP proxy you may, as a special bonus, earn up to 2 additional marks!

2. There may be confidential emails that the 'normal' employees are not supposed to see. Assume that any email that contains the word 'Confidential' in the subject line should be hidden from users who access the mailbox via the proxy. Note that this challenge is rather hard. One way of simplifying it somewhat is to replace the confidential email with some cover email ('Just testing' from a fake email address [perhaps test@example.com] or a message selected from spam that some spam filter recently caught).

3. Insert a line 'Handled by <*username*> where *username* is the proxy username of the individual who downloaded the email. The intention is that, if a user, for example, forwards or prints the email, it may be possible to

determine which user forwarded or printed the email. (Obviously a prudent user will be able to delete such an insertion before forwarding or printing.) However, just seeing the inserted line is sufficient for a mark or two. The fact that the assignment only calls for one proxy username does not matter. Keep in mind that many emails are sent in both text and HTML formats (with `multipart/alternative` MIME header.)

4. Use more than one username on the proxy and only allow one user to DELEte emails.

5. Most importantly: use your own creativity.

9.4 Assessment

This project counts 20 marks. Writing a working proxy server as set out in section 9.2 will earn you a mark of 12. Solving one or more of the challenges will increase your mark; with sufficient additional work, a maximum of 20 will be awarded. However, note that fewer marks may be awarded than specified above if your solutions are not completely correct.