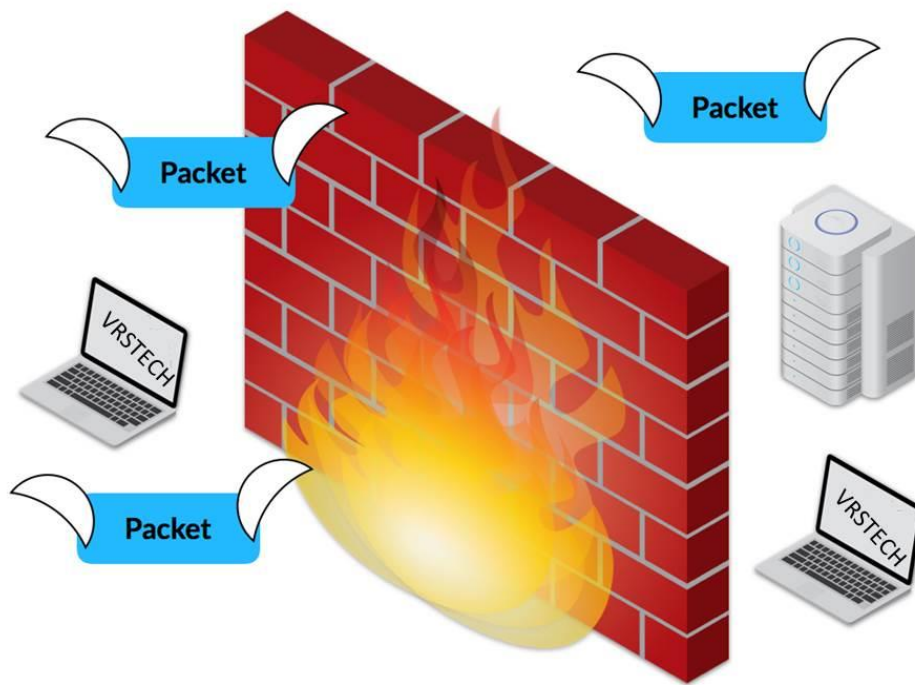

COMP 4108 – COMPUTER SYSTEMS SECURITY

Experience 7 – Network Firewalls



Student Name: Ben Cendana
Student #: 100811212
Date: March 12 2018

Table of Contents

1.0: Introduction	3
1.1: Wireshark.....	3
1.2: Network Protocols	4
2.0: The Network Firewalls	5
2.1: Packet Filters.....	5
2.1.1: Packet Filters Advantages	6
2.1.2: Packet Filters Disadvantages.....	6
2.2: Application Gateways	6
2.2.1: Application Gateways Advantages.....	7
2.2.2: Application Gateways Disadvantages	7
2.3: Circuit Level Gateways	7
2.3.1: Circuit Level Gateways Advantages	7
2.3.2: Circuit Level Gateways disadvantages	8
3.0: Putting It All Together.....	8
3.1: Performing An Network Audit	8
3.2: Which Type Of Firewall Should be Used?	9

1.0: Introduction

Dr. Somyaji provided the class with four foundational papers to read, among them was the paper *Network Firewalls* by Steven M. Bellovin and William R. Cheswick. This experience will document what was learned from that paper and how the knowledge gained from the paper can be used to build effective network firewalls.

The article makes it clear that there are three distinct types of network firewalls these are:

a) Packet Filters b) Application- Level Gateways c) Circuit Level Gateways and each type of network firewall has a distinct advantage and disadvantage.

But in order to use the three network firewalls effectively its important to understand how to read, interpret what type of network traffic is entering/leaving the network, the overall network behaviour and the protocol of each packet.

1.1: Wireshark

The first step in understanding the networks behaviour is to perform a network audit to perform this task a network analyser tool such as Wireshark can be used.

No.	Time	Source	Destination	Protocol	Length	Info
33	3.428939	192.168.2.69	224.0.0.251	MDNS	103	Standard query 0x00a3 PTR _23i
34	4.452619	fe80::426f:2aff:fe01:1c91	ff02::fb	ICMPv6	86	
35	4.453093	fe80::426f:2aff:fe01:1c91	ff02::fb	ICMPv6	86	
36	4.453565	fe80::426f:2aff:fe01:1c91	ff02::fb	ICMPv6	86	
37	4.453788	fe80::426f:2aff:fe01:1c91	ff02::fb	MDNS	153	uestion PTR _friendly._sub._bp2p._tcp.local, "qu" question PTR _ir
38	4.454419	fe80::426f:2aff:fe01:1c91	ff02::fb	MDNS	473	13AE6960CD352E9FA3.local SRV 0 0 46491 09017BF3443213AE6960CD352E9
39	4.454814	fe80::426f:2aff:fe01:1c91	ff02::fb	MDNS	473	13AE6960CD352E9FA3.local SRV 0 0 46491 09017BF3443213AE6960CD352E9
40	4.455176	fe80::426f:2aff:fe01:1c91	ff02::fb	MDNS	473	13AE6960CD352E9FA3.local SRV 0 0 46491 09017BF3443213AE6960CD352E9
41	4.455377	fe80::426f:2aff:fe01:1c91	ff02::fb	MDNS	153	uestion PTR _friendly._sub._bp2p._tcp.local, "qm" question PTR _ir
42	4.455594	192.168.2.69	224.0.0.251	MDNS	103	Standard query 0x00a3 PTR _23i
43	4.455818	192.168.2.104	239.255.255.250	SSDP	216	
44	4.456027	LiteonTe_74:77:d9	Broadcast	ARP	60	
45	4.655067	fe80::9898:5a67:d56d:bf8e	ff02::1:3	LLMNR	95	
46	4.655860	192.168.2.105	224.0.0.252	LLMNR	75	
47	4.757591	fe80::426f:2aff:fe01:1c91	ff02::fb	ICMPv6	86	
48	4.962268	fe80::c09e:8753:6e19:3c2d	ff02::1:3	LLMNR	84	
49	5.067605	LiteonTe_74:77:d9	Broadcast	ARP	60	
50	5.167055	192.168.2.74	224.0.0.251	IGMPv2	60	
51	5.167296	192.168.2.104	192.168.2.255	NBNS	92	
52	5.167298	fe80::c09e:8753:6e19:3c2d	ff02::1:3	LLMNR	84	
53	5.167741	192.168.2.104	224.0.0.252	LLMNR	64	
54	5.167742	fe80::c09e:8753:6e19:3c2d	ff02::1:3	LLMNR	84	
55	5.168188	192.168.2.104	192.168.2.255	NBNS	92	
56	5.168189	192.168.2.104	224.0.0.252	LLMNR	64	
57	5.168190	192.168.2.104	192.168.2.255	NBNS	92	
58	5.168578	192.168.2.104	224.0.0.252	LLMNR	64	
59	5.372416	fe80::c09e:8753:6e19:3c2d	ff02::1:3	LLMNR	84	

Frame 1: 345 bytes on wire (2760 bits), 345 bytes captured (2760 bits) on interface 0
Ethernet II, Src: IntelCor_2d:09:24 (F4:06:69:2d:09:24), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
User Datagram Protocol, Src Port: 68, Dst Port: 67
Bootstrap Protocol (Request)

Figure 1: Wireshark Capture

As we can see in Figure 1 the Wireshark capture provides information regarding the time, the source IP, destination IP, protocol and general info about the packets.

Based on the protocol we can then determine what the network traffic is and to a lesser extent if its malicious.

1.2: Network Protocols

Knowing each protocol and what type of network traffic is associated with that type of protocol can give us clues what that packets primary function is. If we know what the packet is trying to do we can make an inform decision if that packet is malicious or not.

Its almost impossible to know every protocol but by knowing where the protocol belongs according to the OSI Model or TCP/IP Protocol Architecture Model we can understanding what it is trying to do.

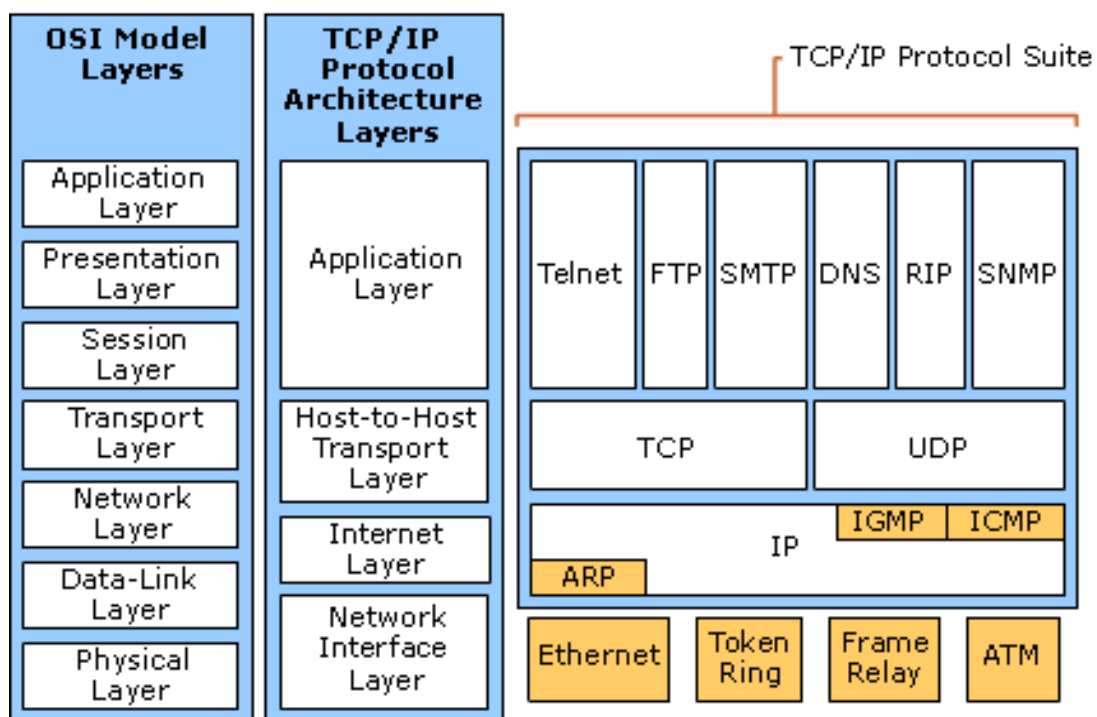


Figure 2: The TCP/ IP Protocols and The Corresponding Protocols Of Each Layer

Each of the layers performs the functions as listed below:

Application Layer: Provides services to lower level layers

Host-to-host Transport Layer: Provides Network Communication

Internet Layer: Datagram Transportation

Network Interface Layer: Network Hardware

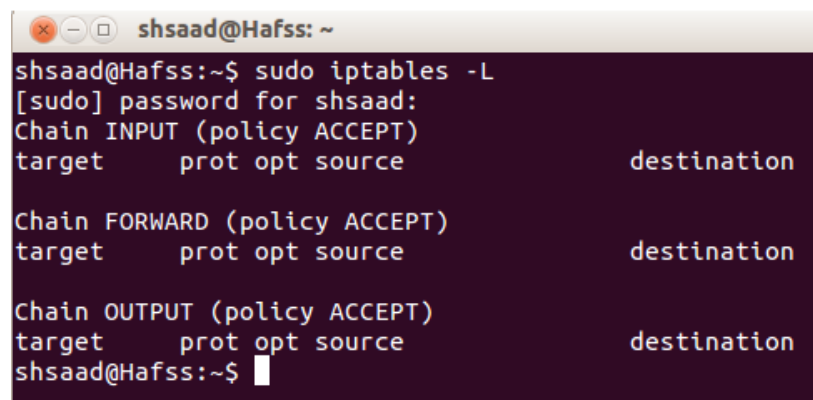
2.0: The Network Firewalls

2.1: Packet Filters

As the name suggests packet filters are firewalls that filter incoming packets based on predefined rules. These rules are normally based on any of the following or a combination of them:

- 1) Source IP
- 2) Destination IP
- 3) Source IP
- 4) Source Port
- 5) Destination Port.

Provided with the Linux operating system is a packet filter called IP Tables which can be accessed by the command **iptables -L**.

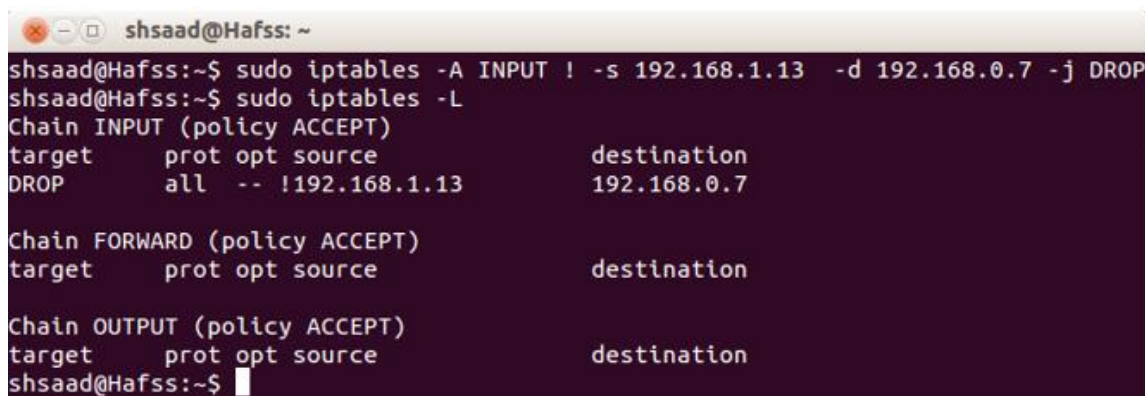


```
shsaad@Hafss:~$ sudo iptables -L
[sudo] password for shsaad:
Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
shsaad@Hafss:~$
```

Figure 3: IP Tables



```
shsaad@Hafss:~$ sudo iptables -A INPUT ! -s 192.168.1.13 -d 192.168.0.7 -j DROP
shsaad@Hafss:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination
DROP       all  --  !192.168.1.13         192.168.0.7

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
shsaad@Hafss:~$
```

Figure 4: IP Tables With A Blocked IP

As we can see in Figure 4 all packets from the source IP 192.168.1.13 are blocked from entering the destination IP 192.168.0.7.

2.1.1: Packet Filters Advantages

The main advantage to using a packet filter is the speed in which it processes incoming/outgoing packets this is because packet filters operate on the transport and network layers.

2.1.2: Packet Filters Disadvantages

The primary disadvantage as highlighted in the article is a dedicated attacker could simply spoof there source IP and enter the network by spoofing or the administrator will detect the intruder but will be forced to create a new rule blocking another source IP.

2.2: Application Gateways

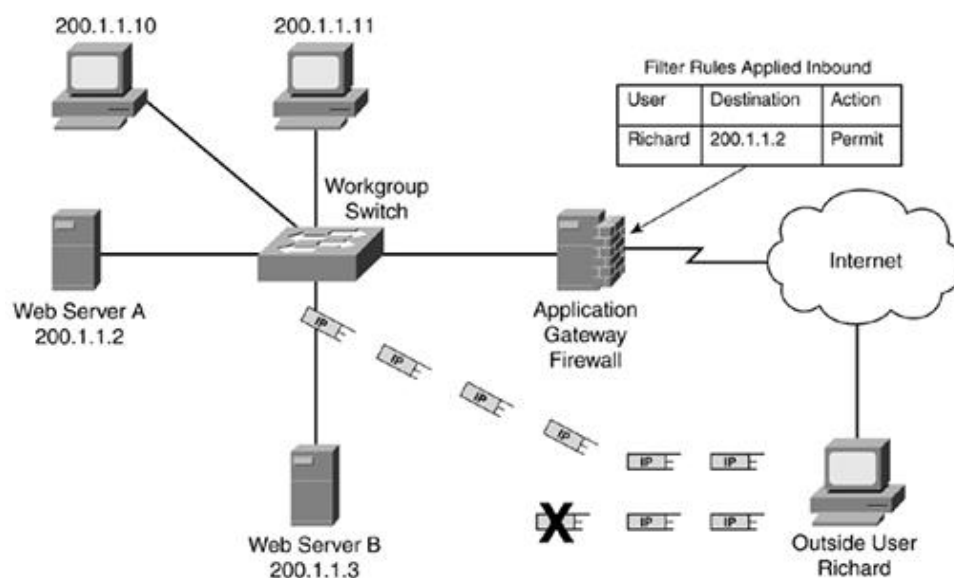


Figure 5: An Example Of An Application Gateway

In contrast to packet filters an application gateway works by having a layer of authentication that checks to ensure anyone accessing the network is valid. For that reason Application Gateways are popular among email and webservers were the user requesting the service can be authenticated.

Typical methods of authentication involve having a separate browser with logins similar to an SSH connection or the client can be authenticated when it sends a TCP request to access the network.

2.2.1: Application Gateways Advantages

The main advantage of this type of firewall is users are authenticated rather than the device which makes spoofing an IP extremely difficult.

2.2.2: Application Gateways Disadvantages

Due to the authentication process, application gateways are slower and offer very few services. This is because this type of firewall requires more processing power and all the processing power will be dedicated to authentication.

2.3: Circuit Level Gateways

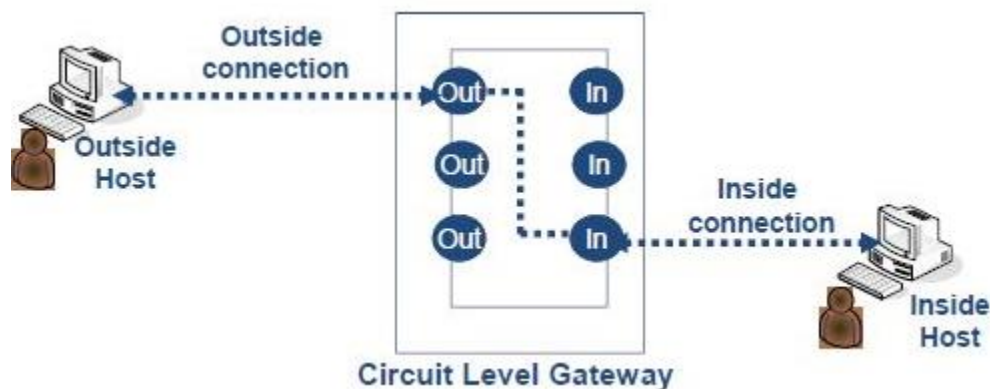


Figure 6: An Example Of A Circuit Gateway

Circuit level gateways operate by checking the connections between the host and the server, if a successful handshake occurs such as a TLS handshake then it knows the connection is secure and should therefore allow the packet to enter the network.

2.3.1: Circuit Level Gateways Advantages

Since sniffers and network topology scanners are not secure connections and do not offer a TLS handshake it can be difficult for an attacker to test the network for vulnerabilities.

2.3.2: Circuit Level Gateways disadvantages

Since most of the emphasis is concerned with building a secure connection, circuit level gateways do not sanitise incoming packets and there contents. Therefore a connection maybe secure but the packets may not be.

3.0: Putting It All Together

3.1: Performing An Network Audit

Putting together what we know about protocols and how to use Wireshark we can determine the network behaviour in Figure 3. We can tell that the TCP protocol was in use since the TCP protocols ar used to initiate communication. We can also tell the client at IP 192.168.2.94 was attempting to communicate to the server at 40.117.145.132.

No.	Time	Source	Destination	Protocol	Length	Info
600	94.733124	40.117.145.132	192.168.2.94	TCP	66	443 → 54161 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1440 WS=256 SACK_PERM=1
601	94.733351	192.168.2.94	40.117.145.132	TCP	54	54161 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0
602	94.734157	192.168.2.94	40.117.145.132	TLShv1.2	277	Client Hello
603	94.768048	40.117.145.132	192.168.2.94	TCP	66	443 → 54162 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1440 WS=256 SACK_PERM=1
604	94.768264	192.168.2.94	40.117.145.132	TCP	54	54162 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0
605	94.769003	192.168.2.94	40.117.145.132	TLShv1.2	277	Client Hello
606	94.825777	40.117.145.132	192.168.2.94	TCP	1514	443 → 54161 [ACK] Seq=1 Ack=224 Win=131328 Len=1460 [TCP segment of a reassembled PDU]
607	94.825780	40.117.145.132	192.168.2.94	TCP	1514	443 → 54161 [ACK] Seq=1461 Ack=224 Win=131328 Len=1460 [TCP segment of a reassembled PDU]
608	94.825784	40.117.145.132	192.168.2.94	TCP	1514	443 → 54161 [ACK] Seq=2921 Ack=224 Win=131328 Len=1460 [TCP segment of a reassembled PDU]
609	94.825785	40.117.145.132	192.168.2.94	TLShv1.2	1223	Server Hello, Certificate, Certificate Status, Server Key Exchange, Server Hello Done
610	94.825786	40.117.145.132	192.168.2.94	TLShv1.2	1514	[TCP Previous Segment not captured] Ignored Unknown Record
611	94.825788	40.117.145.132	192.168.2.94	TCP	1514	[TCP out-of-order] 443 → 54162 [ACK] Seq=1 Ack=224 Win=131328 Len=1460
612	94.825789	40.117.145.132	192.168.2.94	TCP	1514	443 → 54162 [ACK] Seq=2921 Ack=224 Win=131328 Len=1460
613	94.825790	40.117.145.132	192.168.2.94	TLShv1.2	1223	Ignored Unknown Record
614	94.826520	192.168.2.94	40.117.145.132	TCP	66	[TCP Dup ACK 604#1] 54162 → 443 [ACK] Seq=224 Ack=1 Win=262144 Len=0 SLC=1461 SRE=2921
615	94.827125	192.168.2.94	40.117.145.132	TCP	54	54162 → 443 [ACK] Seq=224 Ack=2921 Win=262144 Len=0
616	94.827574	192.168.2.94	40.117.145.132	TCP	54	54161 → 443 [ACK] Seq=224 Ack=5550 Win=262144 Len=0
617	94.827713	192.168.2.94	40.117.145.132	TCP	54	54162 → 443 [ACK] Seq=224 Ack=5550 Win=262144 Len=0
618	94.841374	192.168.2.94	40.117.145.132	TLShv1.2	268	Client Key Exchange, change cipher Spec, Encrypted Handshake Message
619	94.841376	192.168.2.94	40.117.145.132	TLShv1.2	268	Client Key Exchange, change cipher Spec, Encrypted Handshake Message
620	94.916569	40.117.145.132	192.168.2.94	TLShv1.2	161	change cipher Spec, Encrypted Handshake Message
621	94.916653	192.168.2.94	40.117.145.132	TCP	54	54162 → 443 [ACK] Seq=438 Ack=5657 Win=261888 Len=0
622	94.917185	192.168.2.94	40.117.145.132	TLShv1.2	635	Application Data
623	94.917304	192.168.2.94	40.117.145.132	TCP	1494	54162 → 443 [ACK] Seq=1019 Ack=5657 Win=261888 Len=1440 [TCP segment of a reassembled PDU]
624	94.917315	192.168.2.94	40.117.145.132	TLShv1.2	171	Application Data
625	94.920638	40.117.145.132	192.168.2.94	TLShv1.2	161	change cipher Spec, Encrypted Handshake Message
626	94.920719	192.168.2.94	40.117.145.132	TCP	54	54161 → 443 [ACK] Seq=438 Ack=5657 Win=261888 Len=0

Frame 622: 635 bytes on wire (5080 bits), 635 bytes captured (5080 bits) on interface 0
Ethernet II, Src: IntelCom_A0:8e:53 (68:17:29:a0:8e:53), Dst: Fortinet_45:07:3e (08:5b:0e:45:07:3e)
Internet Protocol Version 4, Src: 192.168.2.94, Dst: 40.117.145.132
Transmission Control Protocol, Src Port: 54162, Dst Port: 443, Seq: 438, Ack: 5657, Len: 581
Secure Sockets Layer

Figure 7: TCP ACK Sequence Captured

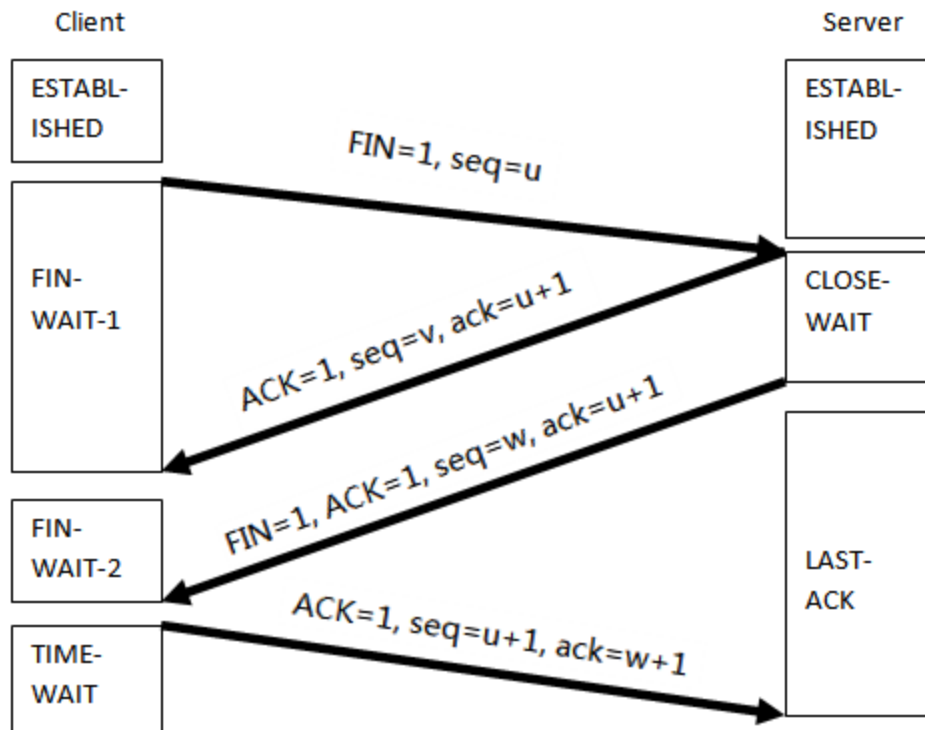


Figure 8: TCP ACK Sequence

The communication between the client at 192.168.2.94 and the server is not much different then the communication modeled in Figure 4. If we look closely at Figure 3 we can tell that the packets are ACK packets indicating that the client was successful *Acknowledged* by the server.

However, an error occurred which was highlighted indicating that the packets being sent back to the client arrived out of order and the client failed to receive the last ack packet from the server.

Based on this information we can get some idea of what kind of network behaviour is occurring, what type of firewall should be used and is the existing firewall working properly?

3.2: Which Type Of Firewall Should be Used?

In all likelihood a packet filter was most likely the cause of the ACK sequence failing as it can block the client/server from receiving packets and establishing some form of communication.

For the example listed above all the firewalls should be able to work fine. But depending on what type of service your network provides the decision should be based on the pros/cons described in section 2.0.

The best type of defense is essential any firewall with every packet being blocked. A dedicated administrator should then be present to perform an network audit and once the checks are completed allow the incoming packet.

