System Administration 1 - 1DV031

Examination 1 - Report

Melina Cirverius
mc222ap@student.lnu.se
Linneaus University
2019-05-12

Introduction

What happens when you use Telnet to connect to a router? Why does TCP need to shake hands? And what exactly is DNSSec?

All of these questions and more is what I will try to answer in this report where I analyse eight different files capturing what is happening in some given network communications. These capture files range from a “simple” request for an IP address, to a secure connection to a server using SSH key pairs. In these capture files I have tried to understand what is happening between the communicating parties by analysing the information given from the different protocols being used to transfer data and information.

The purpose of this report is to show an understanding of how communication can work between for example clients and servers on a network, and how this communication is happening on more levels than might be easily understand by the user. I try to show this by the use of the OSI model which helps explain the different levels the information is being shared on as well as what protocols are being used on these levels.

Findings

File 0

A computer is sending a request to a server asking for the IP address for the host address arla.se. The server then responds with the IP address of the requested host. After this, the computer can communicate directly with the host, which it does by sending ping requests to this IP address and gets replies to each of them confirming that the host received the ping requests.

In the first two packets, DNS (Domain Name System) packets are being sent for the request and reply of the IP address for arla.se. This DNS request is coming from the source with the IPv4 address 194.47.182.46 and is being sent to a destination with the IPv4 address 194.47.110.97 to get a response.
The response is then sent from the previous destination IP back to the original source, that is now marked as the new destination with the response that the requested host address has the IP address 217.114.93.17

In the next package of the file, the original source with the IP address 194.47.182.46 now sends an Internet Control Message Protocol (ICMP) request where it pings the IP address that it got for the host arla.se to se if it responds. It is using the ping type 8 which stands for echo and expects a response.

OSI Model Layer	Protocols used	Sender	Receiver
Application Layer	DNS (RFC 104)		Host: arla.se
Presentation Layer			
Session Layer			
Transport Layer	UDP (RFC 768)	Port: 13823	Port: 53
Network Layer	ICMP (RFC 792), IPv4 (RFC 791)	Client: 194.47.182.46	Server: 194.47.110.97, Host: 217.114.93.17
Data Link Layer		MAC: 24:77:03:e3:3d:58	MAC: 00:24:f9:e1:14:00
Physical Layer			

In this file a total of:
6 packets is being exchanged
640 bytes of data is sent.
and the average size of the packets is 170 bytes.

File 1

First three packets shows the TCP handshake. First the client sends a SYN packet to server to begin the session. The server then responds with SYN ACK and the client responds again with an ACK. After these three packets are sent the client and server are ready to start communicating.

The rest of the scenario is File Transfer Protocol (FTP) communication between the client and the server. The server sends the client a message both to show that the server is ready for the login details of the client, but also to show some rules about using this specific FTP server. This message is divided over several packets and the client then sends TCP ACK packets back to acknowledge that they have received each of the server's response packets.

After this, the client sends the username of “Anonymous” and also the password of “thomas.ivarsson@lnu.se". In this packet there is no use of a secure protocol providing any sort of encryption, so the password is visible in the file.

Next, the client ask the server which operating system it has, with a SYST command, and gets the answer of Unix. After this, the client uses several commands to navigate through the directory to find a specific file. In the end, the file 1978.exe is downloaded and the connection to the FTP server is finished.

OSI Model Layer	Protocols used	Sender	Receiver
Application Layer	FTP (RFC 959)		
Presentation Layer			
Session Layer			
Transport Layer	TCP (RFC 793)	Port: 21	Port: 34308
Network Layer		Client: 194.47.182.46	Server: 204.76.241.31
Data Link Layer		MAC: 24:77:03:e3:3d:58	MAC: 00:24:f9:e1:14:00
Physical Layer			

In this file a total of:
273 packets is being exchanged
336676 bytes of data is sent.
and the average size of the packets is 1233 bytes.

File 2

The client asks for an IP address for the host examinations-test-tivarsson.c9users.io and after receiving this establishes a TCP handshake with the host server and sends a HTTP request to get the page/file hello.html

The server sends an ACK packet that it has received the request and then presents the requested page with the hello world text.
The connection is then finished by the client and host exchanging FIN, ACK packages to end the session in a secure way.

OSI Model Layer	Protocols used	Sender	Receiver
Application Layer	DNS (RFC 104), HTTP		Host: examinations-test-tivarsson.c9users.io
Presentation Layer			
Session Layer			
Transport Layer	TCP (RFC 793)	Port: 44712	Port: 80
Network Layer		Client: 192.168.1.2	Server: 192.168.1.1, Host: 192.158.30.16
Data Link Layer		MAC: 60:6c:66:1e:cf:d5	MAC: e8:08:8b:5a:df:82
Physical Layer			

In this file a total of:
12 packets is being exchanged
1703 bytes of data is sent.
and the average size of the packets is 142 bytes.

File 3

In this file a router on a private network sends out a broadcast to every other instance that is connected on the same network. It does this by using a Routing Information Protocol (RIPv1) that sends a request message about every 30 seconds to see what it gets as a response. The second package in this file is sent after only 10 seconds and in the response it has updated its routing table with a third IP address that it reached. The RIP tries to see how many hops it has to go through to reach its wanted destination to be able to figure out what the quickest route to the destination is.

OSI Model Layer	Protocols used	Sender	Receiver
Application Layer			
Presentation Layer			
Session Layer			
Transport Layer	UDP (RFC 768)	Source Port: 520	Destination Port: 520
Network Layer	RIPv1 (RFC 1058)	Source: 192.168.10.1	Destination: 255.255.255.255
Data Link Layer		MAC: 00:1e:79:f8:1a:41	MAC: ff:ff:ff:ff:ff:ff
Physical Layer			

In this file a total of:
7 packets is being exchanged
722 bytes of data is sent.
and the average size of the packets is 103 bytes.

File 4

File number 4 starts with the client sending a DNS packet to find the IP address of the host adress super-computer.example.com
The client then receives a ICMP packet from a local network saying that the destination is unreachable, and the host could not be found. The request is sent two more times and each time the network responds with the same message saying that the host could not be found.

OSI Model Layer	Protocols used	Sender	Receiver
Application Layer	DNS (RFC 104)		Host: super-computer.example.com
Presentation Layer			
Session Layer			
Transport Layer	UDP (RFC 768)		
Network Layer	ICMP (RFC 792)	Source: 192.168.10.92	Source: 10.254.43.199

OSI Model Layer	Protocols used	Sender	Receiver
Data Link Layer		MAC: e0:db:55:d0:c6:27	MAC: 00:1e:79:f8:1a:41
Physical Layer			

In this file a total of:
6 packets is being exchanged
501 bytes of data is sent.
and the average size of the packets is 84 bytes.

File 5

A computer want to know the destination of a specific IP address in the same private network so it sends an Adress Resolution Protocol (ARP) request asking where this is located. The requested IP address then responds with its MAC address.
The question and answer then repeats.

OSI Model Layer	Protocols used	Sender	Receiver
Application Layer			
Presentation Layer			
Session Layer			
Transport Layer			
Network Layer		Sender: 192.168.10.92	Target: 192.168.10.1
Data Link Layer	ARP (RFC 826)	Sender MAC address: e0:db:55:d0:c6:27	Target MAC: 00:00:00:00:00:00, Response MAC: 00:1e:79:f8:1a:41
Physical Layer			

In this file a total of:
4 packets is being exchanged
204 bytes of data is sent.
and the average size of the packets is 51 bytes.

File 6

In this file there is Telnet communication going on in a private network and this starts with a TCP handshake. We can tell that it is happening in a private network since both the source and the destination IP addresses start with 192.168, and IPv4 addresses in this range are reserved for private networks.

What is happening here is that a client is contacting a router on the network and access this by giving the password. To then access a more secure layer of the router where the client can apply changes and configurations a second password is given to the router. After this the client makes no changes, but instead terminates the connection.

By following the TCP stream via WireShark we can clearly see and read the two different passwords that are used to access the router, showing that the telnet communication is not secure.

OSI Model Layer	Protocols used	Sender	Receiver
Application Layer	Telnet (RFC		
Presentation Layer			
Session Layer			
Transport Layer	TCP (RFC 793)	Source Port: 45132	Destination Port: 23
Network Layer		Source: 192.168.10.92	Destination: 192.168.10.1
Data Link Layer		MAC: e0:db:55:d0:c6:27	MAC: 00:1e:79:f8:1a:41
Physical Layer			

In this file a total of:
136 packets is being exchanged
7930 bytes of data is sent.
and the average size of the packets is 58 bytes.

File 7

In this file there is encrypted communication happening between a client and a server on a private network. The client is using an SSH Protocol (Secure Shell Protocol) to securely connect to a server. By using SSH, session keys are created and exchanged by the client and server so that they can connect in a secure and encrypted way even on an otherwise insecure network. The key used to connect can not be read in the capture files, meaning that we can not read any sensitive information being exchanged.

OSI Model Layer	Protocols used	Sender	Receiver
Application Layer	SSHv2		
Presentation Layer			
Session Layer			
Transport Layer	TCP (RFC 793)	Source Port: 52086	Destination Port: 22
Network Layer		Source: 10.6.13.101	Destination: 10.6.2.128
Data Link Layer			
Physical Layer			

In this file a total of:
80 packets is being exchanged
12062 bytes of data is sent.
and the average size of the packets is 151 bytes.

Methods

To read and analyse the capture files in this report I used Wireshark. This helped me to see the communication happening in the files as well as provided me with all the information regarding the communicated parties and information that was being shared. The main functions I used in Wireshark to help me find the relevant information was the description and information under each protocol being used as well as being able to follow the TCP stream when applicable and bring up the capture file properties. The TCP stream was very useful when it came to getting a clearer picture of exactly what information was being shared between the sender and receiver, as well as to see if sensitive information was available for me to read from the files. This is how I found the files in which usernames and/or passwords were not secure and clearly visible. The capture file properties in Wireshark was what I used to get the information regarding how many packets were being shared in each file as well as the size and average size in these.

I decided to use tables for each capture file to show what protocols were used in each file and what layer of the OSI model this was happening on. The tables provides a clear way of presenting this information as well as the communicating parties and RFCs for the protocols, making this easy to find.

For references to help understand what was happening in the different files, as well as understanding the different protocols I used the course lectures available for System Administration 1 (1DV031) as well as the following literature:
Practical Packet Analysis by Chris Sanders, 2007
Datorn i världen, världen i datorn by Hans Lunell, 2016
Datakommunikation - en inledande översikt by Maria Kihl, 2016

Questions

The TCP/IP–model

The TCP/IP–model is a model of a collection of protocols and how they are used when different servers and computers are communicating over networks. This model is a shorter version of the OSI model as it only contains four different layers, instead of the seven layers of the OSI model. The TCP/IP–model is used to describe how information and data is sent between computers and servers by mapping out the different protocols that are active on the different layers. The use of the model is useful to get a better understanding of what is happening when different parties communicate over networks and how information is being shared in more ways than what may be clearly visible for a client. The four layers present in this model are: the network access layer, the internet layer, the transport layer and the application layer. The reason for learning the TCP/IP-model and why is especially important when it comes to setting up servers and other communicating parties on a network, local or global. An understanding of what information is being shared and how it is shared is needed to be able to set up connections both correctly and securely.

The difference between TCP and UDP

Transmission Control Proctocol (TCP) and User Datagram Protocol (UDP) are both protocols that are used specifically to transfer data packets on networks.
The main difference between TCP and UDP is the safety level of the transmission as well as the speed of sending the data. TCP is a much safer and more reliable way to send data over networks as with TCP there is a guarantee of the data reaching its destination. This guarantee is created by the transmission exchange starting with what is called the three way handshake, which establishes the connection between the two parties that want to send data to each other. TCP also breaks up the data in smaller segments and will put them in the correct order at their destination in case they arrive in the wrong order.
The difference when it comes to UDP is that this protocol is so called connectionless and has no guarantee that the data will arrive at its destination. The UDP instead lets the application protocols take care of both checking for errors and making sure that the data arrives at its destination. The advantage of using UDP to send data is that it is faster than TCP, and can therefor be usable in transmissions where it does not matter if some data packets do not arrive. Examples of when UDP can be used are in online gaming, different sort of broadcasts and live video transfers.

HTTP vs. HTTPS

Hypertext Transfer Protocol (HTTP) is the protocol for transferring data between web servers and clients. HTTPS is basically the more secure version of HTTP, the ’s’ at the end stands for ‘secure’ and was created since the data sent via HTTP is not secure as the data is not encrypted. What the extra ’s’ in HTTPS really means is that the website uses Transport Layer Security (TLS), or previously a Secure Socket Layer (SSL), to send data between the client and site in a secure way, whereas a website using HTTP sends data that can clearly be read by a third party.

SSL and TLS are protocols whose purpose is to encrypt data being sent over networks so that is secure from attacks by outside parties that should not be able to access the information. This is used so that clients can be ensured of a secure way of providing their information to for example different websites. SSL was replaced by TLS as this was found to be much more secure and in a way an upgraded version of the original. By using a TLS protocol the data is usually encrypted especially for each session to ensure the best security in the information being sent. This might for example be very important for websites or other applications where a user is asked to provided sensitive information such as passwords or even addresses and payment information.

The SSL and TLS protocols are used for different forms of communication happening between clients and servers, and client to client on for example website, messaging services such as email or direct message applications and even applications that has a function for voice calls. The protocols are also used between servers and the web browsers that a websites presents its information on, the provide a secure way for this information to get from the server to the end user.

DNSSec

DNSSec or Domain Name System Security Extensions are extensions the the regular Domain Name System (DNS) that are being used to make sure that the DNS data being received is the same data that was originally sent, and have therefor not been manipulated or changed in any way from other parties. What DNSSec does it that it gives a digital signature to the data being sent, this signature can then be checked to make sure that the data is the same as what was provided by the DNS zone owner. DNSSec is not used to encrypt the data in any way, but is instead used for the purpose of showing that the data has not been changed on the way to the receiver as well as showing the receiver that the data was sent from the correct zone owner.

A DNS zone has both private and public keys, the public keys are what is used by the receiving end to make sure that the data being sent was sent from the expected owner. A resolver, which is what receives the information first and checks the digital signature with the zone owner, will discard the data if the signature is not correct. If this is the case the data will not be received by the end user, but they will instead receive an error message.

Network Segmentation

Network segmentation is a way of creating subnetworks from a computer network, where each segment of the network becomes a subnetwork. There are several advantages of doing this as it both provides higher security on the network and better performance.

In ways of providing higher security this is for example that when using multiple subnetworks a firewall can be put up separating these from each other, in that way if some data is sent to one subnetwork it can be scanned there to see if there is anything in it that can be harmful to the network or system before it is sent forward to the other side of the firewall. Another way network segmentation helps with security is that it can contain the local network from being visible from the outside, and the local communication is happening on a closed part of the network only accessible to parties on that local network. In the same way it can limit outside communication and data transfer to some subnetworks, instead of on the whole network, for example allowing

visitors access only to chosen parts. Network segmentation can also help by limiting something that is going wrong on the network to one part or segment of it, and stopping it from spreading to other parts.
This method of dividing up the network can also help in improving the performance on the network, for example by also dividing the workload between the different subnetworks. Data can be chosen to be routed in ways that will be the most efficient and for example might only go through the network by a few of the subnetworks instead of being sent to all parts of it. This way of dividing up the workload and how the information is being sent can reduce the traffic on the local network and making it better in its performance.

Conclusion

The work that I have presented in this report have taught me a lot when it comes to the different protocols being used to communicate and send data between different parties over a network. It has been interesting to analyse the capture files and to try to understand exactly what is happening in the different protocols present in each file. I found that some files were easier to understand than others, and also that sometimes I felt that I could explain the actual user scenario that was happening, whereas in other files I can really just explain the technical bit of what was being sent back and forth without really being sure what the user might be doing, or if there even was a user present.

When starting to look through all the information available in Wireshark I found that I sometimes started to dig down a bit too deep, trying to understand every single little bit of information being provided, even though I then understood that some of it was a bit too much for me to get into and comprehend. It was fascinating and a learning experience to go through the different capture files and feel that the more I looked through them, used different options available in Wireshark, and read up on the course literature and lecture notes, the more I started to understand what was happening in each file.

What I take away from working on this is that it is really important to know in what was and through which protocols data and information is being sent on networks, especially when it then comes to setting up connections to servers and sending information. Seeing protocols such as telnet where to data such as passwords were clearly visible in the files, really showed the importance of a secure and encrypted connection and how easily information can be gathered by outside parties otherwise.