**Jane Aldakkour**

**Wireshark Project**

**Network Management and Information Security**

**CDIM 6340**

In this assignment you will explore the use of Wireshark to detect networks, capture packets, and analyze traffic using the Coursera course on Wireshark that I assigned to you. The course runs for about 2 hours. Perform the course.

Your write up again uses the 3-question format: what did you do, what were the results, what did you learn. Be sure to tell me how Wireshark works and how the results are used to inform security planning and risk assessment.

To do the write up, use the 3-question format:

1. What did you do – in this section tell me how you did your network and protocol identification and packet capturing how the software works, and how you analyzed the output. This section is all about how you are doing things, do not put results or outcomes here, just how you did it.

I started this project by going to Coursera website, as instructed, and following the instructions there to complete the assigned lesson. In doing this, the instructor walked through the steps on how to use Wireshark to capture packets running through the network. Wireshark is a network analysis tool that captures packets in real time and presents them in a way that is readable to people. It can also capture packets that were already saved for analysis. Wireshark shows the user all kinds of information, such as the time it took to transmit, the source, the destination, the protocol, the length, and the information in the packet. Specific ports can be chosen for monitoring, or the user can monitor the entire network. Packets of information can also be filtered by many things by applying a “display filter”. Wireshark will only allow you to apply valid filters and will provide suggestions if an incorrect filter option is typed into the filter bar.

The first assignment in this course was an overview of Wireshark and how it works. The instructor walked through how to analyze already captured packets and how to capture and analyze packets in real time. To start, the instructor had me open the RADIUS.pcap packet so that she could demonstrate how to analyze saved packet captures. Upon opening the packet, it showed several things, such as the time that the packet transfer took, the source from which it came from, the destination, the protocol, length, and information. Below this information, there was information about the contents of the packet. After this was shown, we closed the file and when back to the “home page” of Wireshark, so that she could demonstrate how to capture packet transfers. On the home page, there is a section labeled “Capture,” underneath which one can see the different machines that are connected to the network. In this example, there are two – Ethernet 3 and Adapter for loopback traffic capture. We selected Ethernet 3 to analyze, which took us to the capturing page which showed all the packets being sent on the network in real time, with all the information that was in the saved packet capture described above (the source, the destination, the protocol, etc). There were a lot of packets, so the next thing I learned how to do was how to filter the packet traffic to show only what I want to see. The instructor used the DNS filter as an example for how this could be done, by typing “DNS” in the bar at the top of the page. In her example, there was no DNS traffic, so she generated some by going to the command center and pinging coursera.com. She then showed how the filter only picked up on that ping to coursera, and in the “Info” column, Wireshark showed that “coursera” was the website that was pinged. To stop the packet capture, the instructor showed us that the red square button in the top left screen was the button to press. Then, going to File and selecting Close returns the user to the home page with the option to save the packet capture or to continue without saving. This concluded the first exercise.

The next exercise Wireshark had me perform was to capture RADIUS traffic. The purpose of this task was to analyze one of the clear text protocols. RADIUS is a client/server networking protocol which operates on a port (1812) that provides centralized authentication, authorization, and accounting management for users who connect and use a network service. To start the exercise, the instructor had me open the RADIUS Server icon, which was saved to the Cloud desktop. This was not working for me, however I was able to still get to RADIUS, as the instructor showed that her icon took her to the website “idblender.com/tools/public-radius.” The instructor had me insert a username and create a password on the website that appeared. She then had me return to Wireshark to start capturing the packets that were being sent. She showed another way to filter packet information, which is to filter from the beginning using the filter on the home page underneath “capture.” This way, the only data that is captured is data relevant to what we were looking for. The instructor had me filter using port 1812, since this is the port that RADIUS operates on. There was no data that was being captured, and so the instructor then had us open the icon “NTRadPing Test Utility” from the desktop, which sends authentication requests to RADIUS server. The instructor had us put in the IP address provided on the RADAIUS server website and change the port to 1812. I then typed in my username and password that I created for RADIUS on the NTRadPing before clicking send. I then reviewed the results on this packet transfer. The instructor then showed how to decrypt encrypted packages using Wireshark. She had me go to edit, then preferences, protocol, then RADIUS. On this page, it asks for the “Shared Secret”, which is the word “secret”, that once typed in the box, decrypts the encrypted message.

The next exercise Wireshark ran through was to analyze one more unencrypted protocol – HTTP. HTTP is an application layer protocol designed for communications between web browsers and web servers that runs on port 80. HTTP is even more unsecure than RADIUS is, as some packages in HTTP do not even encrypt the password like RADIUS does. The instructor has me open HTTP basic on the desktop, which asks for a username and a password. I then went back to Wireshark to enable packet tracking, filtering on port 80, then went back to HTTP basic and typed in the username “username” and password “password.” I then went to Wireshark to stop the packet tracking, then to analyze the data that appeared.

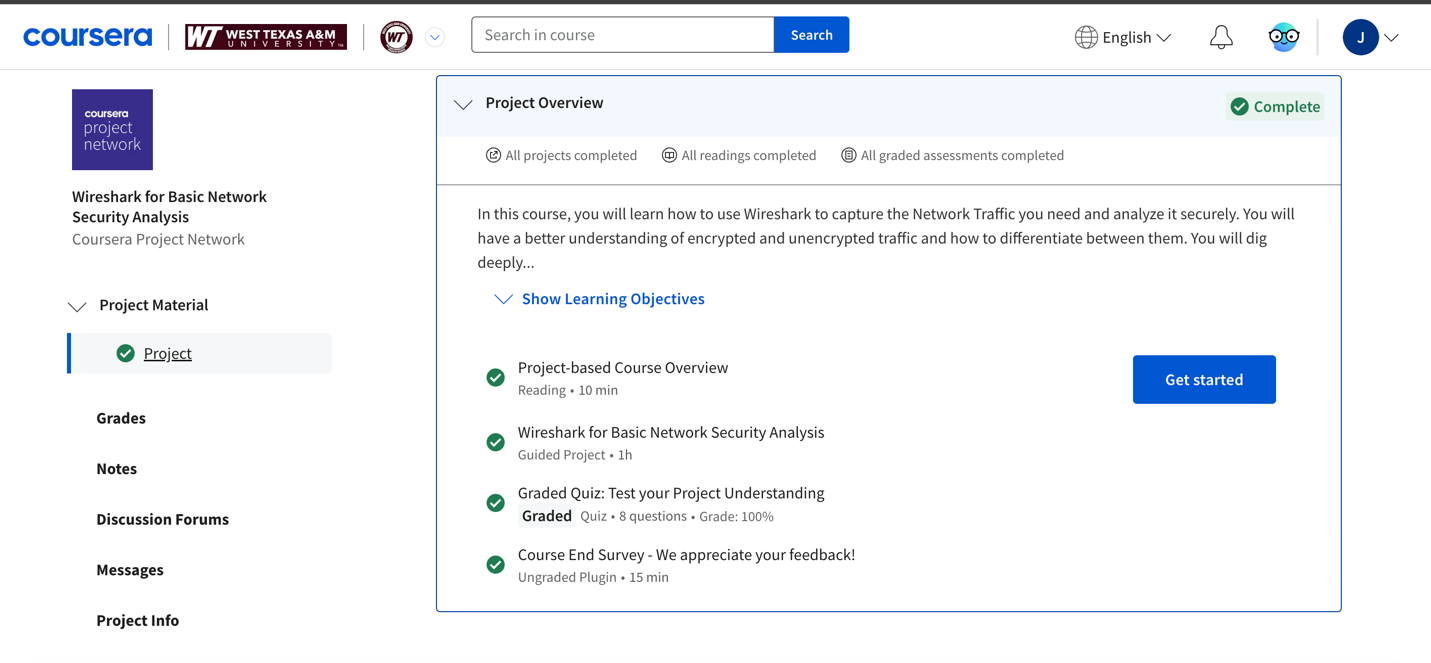
In the next exercise, there were two tasks completed. The first was to capture from a second HTTP example – a form-based authentication which was created to mitigate the downsides of the basic authentication. Form-based authentication uses the standard HTML form fields to pass the username and password values to the server. I opened the icon on the desktop labeled “HTTP form-based" which took me to a site. I then went back to Wireshark to start packet tracking and filtered using port 80, before returning to the website to enter the username “admin” and password “12345.” Once I logged in successfully, I returned to Wireshark to analyze the packet information. The difference between this HTTP example and the previous one is where the credentials were located. In this example, the data is found in the HTML form line instead of the Hypertext Transfer Protocol line. It is not encrypted at all. The main reason that it is used is because it allows a user to be signed out of their account after a certain amount of time without a sign-out button. Once this was completed, the instructor then had me return to the Wireshark homepage to perform the second half of this exercise, which was to capture some DNS packets. I closed the packet traffic tracking for the HTTP exercise, then returned to the home page. I then filtered for DNS traffic only (on port 53). There was no DNS traffic occurring, so per the instructions given, I opened the command prompt and pinged “[www.rhyme.com](http://www.rhyme.com).” I then returned to Wireshark to analyze the captured data.

The next exercise in Wireshark tasked me with running through Telnet, the last unencrypted port that will be looked at. About halfway through this assignment, the Cloud space that I had to work in timed out on me. I attempted to log out and log back into Coursera, thinking it was perhaps something that I had taken too long to respond to while rewatching the instructor’s videos, but that did not work. I then read that I had used up all the time available for the Cloud and would no longer be allowed access but could still watch the instructor's videos. For the remainder of the lessons, that is what I did since I could not access the cloud space to practice in. I watched the Telnet instruction video. The instructor opened the SDP.org icon on the desktop and entered an email and preferred login. She then went back to Wireshark to filter for Telenet packets, located on port 23, and started to track them. There was nothing coming in, and so to generate some Telenet data she went to Windows Powershell (a command line used for task automation and configuration management). She typed in Telenet tty.sdf.org in the command line, it then asked for the login information. After this is done, she returns to Wireshark to stop the packet traffic tracking and to analyze the data. One last thing that she shows is how the Telenet data is not encrypted at all. If one right clicks on the first Telenet packet on the list, then go down to follow, then to TCP stream, it shows everything that was on the Telenet session.

The next Wireshark exercise started walking through the encrypted protocols. This exercise focuses on Telenet (unencrypted) and SSH (encrypted), which are both used to remotely access and manage a device. Telenet uses port 23 and SSH uses port 22. The SSH packet traffic is generated using the same method as it was for Telenet, which is through the Powershell. This time, instead of filtering the packets by port, the instructor filters by host, typing “host tty.sdf.org” to start the packet traffic tracking. She then opened Powershell and ran ssh and the username, responded yes to the question prompted, and then entered the password. After, the instructor went to Wireshark and analyzed the data.

The last exercise Wireshark ran through was to capture data through HTTPS, which is just HTTP over TLS and operates on port 443. The instructor started running a packet traffic track on port 443 using Wireshark. Immediately, there was packets being captured. Since this was the case, she stopped the packet tracking and began analyzing the data. She shows how the data is all encrypted, and how it cannot be unencrypted unless the hacker has the pre-master secret key. The instructor showed how the master-key can be found by going through system preferences, environmental variables, new user variable, then searching for the “sslkeylogfile.” The instructor ran through one last HTTPS test before ending the Wireshark program.

After completing the course, I completed the quiz as instructed (I had to retake it once, as I received a 75% on the first attempt and needed an 80% to pass). I then completed the survey and took a screenshot of the completed course to finish the project.



1. What are the results – in this section tell me about what you see in the networks and protocols identified and the captured packets. Do any packets reflect a cyber-attack? How can you tell? Analyze the attack surface presented by the network and packets. Attack surface is about all the areas that are subject to a cybersecurity attack.

In the second exercise, where I analyzed the RADIUS packet traffic, there were a few things that I noted as possible security risks. All the columns in the packet tracking page show valuable information that hackers could use to find weaknesses. The source and destination IP addresses is dangerous for hackers to have because it shows which devices are communicating with each other. If a hacker can watch this long enough, patterns will be recognized which can be used to exploit the network. There are many cyber-attacks that could be developed this way. For example, if a hacker sees that a person is regularly communicating with a specific website (for this example, I'll use a local gym’s website) then the hacker can use the knowledge that this user goes to the gym, or is looking to go to the gym, to design a phishing attack (such as fake advertisements for that gym). Another attack that a hacker might try is a “man in the middle” or MITM attack, which is where a hacker impersonates both senders and receivers in a network, then intercepts and alters the information being sent. By being able to observe the patterns in packages being sent, this would make it much easier for a hacker to perform an attack such as this. Another reason that this data is dangerous is because in the “info” column, it provides the viewer with some detail about the information that is being sent. In the example that the instructor provided, it shows that there was an “access request” and “access acceptance” between the two devices communicating. This shows that there was an authentication that was happening between these devices, which is not good. A hacker can use this knowledge to target these connections. Furthermore, since the RADIUS data is unencrypted and only uses secret key (shared key) to hide the password only between the server and the client, when one selects the RADIUS protocol drop down on the Wireshark packet trafficking, there is a lot more information that is shown, including the username of the user authenticating. With this information in hand, a hacker could perform many different attacks to figure out the user’s password – one of which being a brute force attack where the hacker uses trial and error to guess the user’s password. One can also use Wireshark to decrypt these kinds of encrypted data.

Another point where I saw a serious security deficiency is through using unencrypted HTTP. In the example that is shown in Coursera, the data that is found in the HTML form line instead of the Hypertext Transfer Protocol line and it is not encrypted at all. Form-based authentication was created to mitigate the downsides of basic authentication. Form-based authentication uses the standard HTML form fields to pass the username and password values to the server. The main reason that it is used is because it allows a user to be signed out of their account after a certain amount of time without a sign-out button. It was created for use on HTTPS, in which the entire packet is encrypted using TLP and encrypting the password was not necessary. There are many websites that operate on HTTP rather than HTTPS, meaning that they are unsecure and not encrypted. Anyone can see the data that is being sent on these packets, which poses a major security risk. In the past, I have not paid much attention to whether websites are operating on HTTP or HTTPS. I have logged on to many websites while operating on public networks (usually at Starbucks working on homework) and therefore could have exposed my username and password to anyone watching packet trafficking on these shared networks.

One of the last possible security risks that I noted from this project was during the last exercise in Coursera. Although this exercise was demonstrating HTTPS secured data transfers, it also explained that all encrypted data has a decryption key (the pre-master key), which is in a file on the device. This is a risk to the security of the network as well, for if a hacker can gain access to this master-key, then they would be able to decrypt the message. The instructor showed how the master-key can be found by going through system preferences, environmental variables, new user variable, then searching for the “sslkeylogfile.” One of the upsides to this is that the process is time-consuming and must be done for every website that the hacker would want to gain access to. The fact that this information can be found at all on the network is still a security concern that should be factored in.

1. What did you learn – discuss your takeaways from the assignment, tell me what you learned about digital networks, packets, and attack surfaces, and how you can use them in the future, also discuss how they could be of value to the organization in the future.

I learned how packet capturing can be used to gather important information that could be detrimental to the user should it fall into the wrong hands. I also learned about the various programs and websites, such as Wireshark, that can be used to capture packet transmissions and the information that is stored in them.

To start, I learned the differences between encrypted data and unencrypted data. Encrypted data is anything protected by an encryption algorithm to prevent unauthorized access to the data being sent, for which there is an encryption key that is used to view translate messages and view them in the original form in which they were sent. Unencrypted data is information sent as it is (and does not have any form of protection). Unencrypted data is unsecure and is available to anyone to view. A common example of encrypted data vs unencrypted data is HTTP and HTTPS. HTTP is the unencrypted version, and HTTPS is the encrypted secure version which ensures that the data is secure and encrypted using transport layer security (TLP).

Some of the biggest takeaways that I had from this assignment were in the discussions regarding how dangerous it is to use unencrypted ports to deliver messages. I had not realized how easy it was for anyone to see this data when it is unencrypted, and how accessible it was as well. In terms of security, there are many types of cyber-attacks that a hacker could perform based solely on the information that they can gather by running Wireshark on a network. As I mentioned earlier, there is a lot of information Wireshark provides to the user, such as the source and destination IP addresses, which devices are communicating with each other. If a hacker can watch this long enough, patterns will be recognized which can be used to exploit the network. There are many cyber-attacks that could be developed this way such as phishing attacks, “man in the middle” or MITM attacks, as well as many others. Information provided in the “info” column opens users up to potentially having their authentication information exposed. As a user on my network, I must be more vigilant moving forward to ensure that websites I provide my authentication information to are encrypted. In the past, I have not paid much attention to whether websites are operating on HTTP or HTTPS. I have logged on to many websites while operating on public networks (usually at Starbucks working on homework) and therefore could have exposed my username and password to anyone watching packet trafficking on these shared networks. Moving forward, I need to be more careful and aware of this – especially when operating on networks outside of my own.

Resources:

<https://projects.coursera.org/run/2BEEVUWWVWGBLYZ5EUGZ>