**Lesson 2.3 – Server Exploits**

**\*\*Instructions:** Please change the text color of your responses to red text. Please organize the endings to each page.

**Activity 2.3.3 – Server Attacks**

1. Why do you think the data in the Packet Bytes pane (the third pane down) is referred to as a “hexdump”? (Step #6)
2. Go to Packet 484 and record the Destination MAC address. (Step #9)
3. Scroll to the top of the Packet list. Review the Info column to determine which packet responded to “Who has 192.168.1.64? Tell 192.168.1.254” with their MAC address. In Packet Details, what is the MAC address of the router who has host 192.168.1.64? (Step #12)
	1. Hint: After you apply the filter, look for Packet list Info that may answer the question. The answer in the Info column is:
	2. Save a screenshot of your Wireshark window.
4. Document the ip.addr Wireshark filter and describe what it does. (Step #14)
5. What are the two IP addresses that performed the handshake? (Step #15 A)
6. How long did the handshake take? (Step #15 B)
7. Document this Wireshark filter and describe what it does. How do the ip.addr and ip.src filters differ? (Step #16)
8. Select the first TLSv1 packet near the top of the Packet List that comes from the host. (Step #17)
	1. Student hint: If students can’t find the first TLS packet coming from host 192.168.1.64, it’s packet 85.
9. Do you think it would be harder to spoof (fake) a TLSv1 handshake than a TCP handshake? Why or why not? (Step #19 A)
10. Near the bottom of the pane, you can see an entry for “Full Request URI”. It’s the web page that was requested. (Step #24)
	1. Save a screenshot of the full URI.
11. What is the purpose of the website? (Step #25)
12. Document this Wireshark modified filter and describe what it does. (Step #26)
13. Explore packets whose Info indicates that they contain text/HTML (as opposed to a gif, png, or undefined data type). (Step #27)
	1. Student hint: If students struggle to find the WebFontConfig script, instruct them to look for the <SCRIPT> and </SCRIPT> tags that encompass the WebFontConfig script name.
14. As you become more of a security expert, you’ll be able to analyze these types of script. For example, you can see that the WebFontConfig script loads another script called webfont.js from a site called ajax.googleapis.com. Using a web browser, is the site a safe and reliable source for embedding scripts in a web page? Why do you think you needed to validate that ajax.googleapis.com is not suspicious or malicious? (Step #28)
15. Show the entire script in the Details pane and save a screenshot of your Wireshark window.
16. How many different log-in attempts were made on this website? Approximately how close together did these packets arrive (in seconds or minutes, not milliseconds)? (Step #35 A)
17. Observe the values for the usernames and passwords. Do you think this data represents a legitimate log-in attempt or a brute force attack? Justify your answer. (Step #35 B)
18. Briefly scan this data. You are looking for failed login attempts that may indicate a brute force attack. Just after the Login header (<H2>Login</H2>) you should see a form and then a response. The response appears after the form content. What is that response? (Step #37)
19. Close the Follow window. All packets for the stream are now showing in your Packet List. When you selected Follow > HTTP Stream data earlier, what Wireshark really did was to create a filter for you. What is the filter? (Step #38)
20. Save a screenshot of your Wireshark window. (Step #38)

CONCLUSION

1. Why would website applications encrypt or encode data that they send over the network? Be sure to include what you learned in Wireshark in your answer.
2. Why would a black-hat hacker want to spoof a MAC address?
3. Suppose a brute force attack was made using a computer algorithm that runs automatically. Describe how the packet capture data for this attack would look different than the data you saw in this activity.