ARL-SR-0468 ● FEB 2023

Hands-on Cybersecurity Studies: SSLStrip Analysis

by Jared Aguayo and Jaime C Acosta

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-SR-0468 ● FEB 2023

Hands-on Cybersecurity Studies: SSLStrip Analysis

Jaime C Acosta
DEVCOM Army Research Laboratory

Jared Aguayo
University of Texas at El Paso

REPORT DOCUMENTATION PAGE

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
		START DATE	END DATE
February 2023	Special Report	August 2022	December 2022

4. TITLE AND SUBTITLE
Hands-on Cybersecurity Studies: SSLStrip Analysis

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**

6. AUTHOR(S)
Jared Aguayo and Jaime C Acosta

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
DEVCOM Army Research Laboratory
ATTN: FCDD-RLA-ND
White Sands Missile Range, NM 88002

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-SR-0468

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES
ORCID IDs: Jared Aguayo, 0000-0003-4802-8590; Jaime C Acosta, 0000-0003-2555-9989

14. ABSTRACT
Network communications are critical for everyday life. It is essential that the technologies involved are secure, as it is difficult for individual users to accurately assess that the information transferred between computing devices stays among the intended recipients. In the web domain, a great deal of trust is placed on the computing machines when accessing remote services. For this reason, digitally signed certificates are used to verify communicating parties are who they say they are and, furthermore, they are used to establish secure, encrypted sessions. However, there are several other potential issues that must be considered, including intermediate network nodes through which communication must flow to reach its destination. For example, these intermediate nodes may be able to modify the communication infrastructure between clients and servers, as documented and demonstrated by Moxie Marlinspike (in *New Tricks for Defeating SSL in Practice*) and a proof-of-concept software tool called SSLStrip. This report describes a hands-on exercise demonstrating these issues and their remediations.

15. SUBJECT TERMS
security awareness; security testing; hands-on cybersecurity; CyberRIG; SSL; SSLStrip; HTTP/HTTPS; IP routing; Network, Cyber, and Computational Sciences

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT	b. ABSTRACT	C. THIS PAGE		
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UU	23

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
Jaime C Acosta	(575) 993-2375

STANDARD FORM 298 (REV. 5/2020)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures iv

1. Introduction 1

1.1 Exercise Overview 1

1.2 HTTPS/HTTP and HSTS 1

1.3 Secure Web Communications 2

2. Setup and Configuration 2

3. Learning Objectives 3

4. Exercise 4

4.1 Activity 1: View Web Browser Network Traffic 4

4.2 Activity 2: HTTP Versus HTTPS 6

4.3 Activity 3: Setting Up MITM Against a Windows Machine 8

4.4 Activity 4: Perform SSL Stripping Attack 12

4.5 Activity 5: Successful SSL Stripping 12

5. Conclusion 14

6. References 15

List of Symbols, Abbreviations, and Acronyms 16

Distribution List 17

List of Figures

Fig. 1 Terminal location in Kali Linux 5
Fig. 2 Wireshark command 5
Fig. 3 Selecting eth0 interface 5
Fig. 4 Add a filter “tls” in Wireshark 6
Fig. 5 Handshake between the client and web server 6
Fig. 6 Restart traffic sniffing button 7
Fig. 7 After clicking restart, select continue without saving 7
Fig. 8 The badssl website 7
Fig. 9 HTTP traffic in Wireshark 8
Fig. 10 Command to get hosts on network 8
Fig. 11 Lists of hosts on the network 9
Fig. 12 Allow packet forwarding 9
Fig. 13 Adding iptable rule to packet forward 9
Fig. 14 Command to check routes 9
Fig. 15 arpspoof command 10
Fig. 16 Display arp table on machine 10
Fig. 17 arp table of machine 11
Fig. 18 Wireshark showing MITM attack 11
Fig. 19 sslstrip command to downgrade HTTPS traffic 12
Fig. 20 View of real time logs of the attack 12
Fig. 21 Vulnerable HTTP login page to test 13
Fig. 22 Wireshark traffic of vulnerable login page 13

1. Introduction

Network communications are critical for everyday life. It is essential that the technologies used for this data exchange are secure, as it is difficult for individual users to accurately assess that the information transferred between computing devices stays among the intended recipients. In the web domain, a great deal of trust is placed on the computing machines when accessing remote services. For this reason, digitally signed certificates are used to verify communicating parties are who they say they are and, furthermore, they are used to establish secure, encrypted channels. However, there are several other potential issues that must be considered, such as the intermediate network nodes through which communication must flow to reach its destination. For example, these intermediate nodes may be able to modify the communication infrastructure between clients and servers, as documented and demonstrated by Moxie Marlinspike through a proof-of-concept software tool called SSLStrip.[1]

This report contains a learning exercise aimed at introducing concepts related to secure web communications and some considerations of which users should be cognizant.

1.1 Exercise Overview

The exercise focuses on the use of a publicly available testing tool to analyze and test secure web network traffic. The exercise demonstrates how users may experiment with the secure transfer of data from websites to clients as well as an insight into man-in-the-middle (MITM) attacks and how they may be identified and mitigated.

Participants walk through a mock scenario to learn about the process of securing connections from clients to web servers when a website's content is accessed. Additionally, participants will test a fictitious network against a MITM attack to gain a hands-on understanding of the importance of the secure transfer of data, especially on websites that request and/or contain sensitive information.

1.2 HTTPS/HTTP and HSTS

Accessing websites over the Internet has inherent security risks. The level of risk is especially substantial when sensitive information is part of the communication, such as credentials or private records. Hypertext transfer protocol (HTTP)[2] is the original communication protocol used to send web content between clients and web servers. However, this protocol is not considered secure, as data is transferred in

clear text; any systems between the source and destination may see the data without any obtrusions. Hypertext transfer protocol secure (HTTPS)[3] adds a layer of security to the communication by using encryption, which makes information unreadable to unintended third parties. This alleviates many security issues; however, other weaknesses still exist. Users need to know when the HTTPS protocol is employed; otherwise, their decision to trust or distrust a connection to a server may be unsubstantiated. This was an issue for many years; web browsers would not indicate effectively if a connection with a remote site was encrypted (i.e., using HTTPS) and because of this software such as SSLStrip would take advantage to deceive users into sending unencrypted data to unintended parties.

Modern browsers have for the most part overcome this issue by employing HTTP Strict Transport Security (HSTS),[4] which only connects to remote sites if they offer an HTTPS connection. Otherwise, the site is blocked and non-accessible to users.

1.3 Secure Web Communications

Secure sockets layer (SSL),[5] now superseded by transport layer security,[6] is the underlying protocol used by HTTPS to enable data confidentiality and integrity. A digital handshake between the client and the web server initiates an encrypted channel between the two communicating parties. The SSLStrip software tool[1] attempts to intercept this handshake and subsequently disable encryption from the client and server, which will then result in all traffic sent as clear text. While there are now mitigations provided by modern browsers, this process is important to understand as it is not limited to web communications.

2. Setup and Configuration

The setup of the exercise includes two virtual machines (VMs) and several open-source software tools. These technologies are listed as follows:

- SSLStrip 1.0[1]
- arp-scan 1.9.8[7]
- arpspoof 2.4b1[8]
- iptables 1.4[9]
- Wireshark 4.0.1[10]
- Kali 2022.1 64-bit VM[11]
- Windows 10 64-bit VM[12]
- VirtualBox 6.1.30 64-bit[13]

The US Army Combat Capabilities Development Command Army Research Laboratory's South Cyber Rapid Innovation Group Collaborative Innovation Testbed is used to host the VMs. The Kali Linux VM is used without any modification, from a standard default installation. The Windows 10 VM is hosted in an unmodified state as well. Both machines are on the same network. Participants set up the testing environment for the scenario.

As part of the exercise, participants become familiar with many publicly available network analysis tools. The "arp-scan" tool is used to show active devices on a network and "arpspoof" is used to impersonate one of the identified addresses. The "iptables" tool enables the control of packet transmissions. The "Wireshark" tool is a network sniffer that allows analysts to view packets as they are received and transmitted from a device; in this case, before and after the MITM attack and the use of SSLStrip.

All artifacts of this learning module are packaged using the repeatable experimentation system.[14]

3. Learning Objectives

The purpose of the exercise is to gain a basic understanding of the protocols that are used to access web pages as well as how a MITM attack works. This will give insight into potential dangers that may arise when using older software and distrusted networks and devices. During the exercise, participants analyze a realistic scenario in a controlled environment by enacting actions from various perspectives. The following general cybersecurity topics are emphasized:

- Ensuring data confidentiality and integrity is crucial to network security.
- HTTPS is now in wide use and modern browsers use HSTS, which allows web interactions to occur only when HTTPS is enabled. HTTPS encrypts data sent over a network, while HTTP sends data in plain text.
- MITM allows a third party to act as proxy and relay information between the client and an application. SSLStrip uses MITM to intercept traffic between a client and a web server and then downgrades the connection to use HTTP instead of HTTPS. This allows a third party to compromise the confidentiality and integrity of the communication.
- The iptables software enforces rules for accepting or rejecting traffic. It is also used for routing traffic to and from specified physical interfaces on a device.

The following learning objectives are associated with the exercise:

- Cyber Security Awareness: Participants will understand the mechanics of SSLStrip and why it no longer works when using modern browsers that enforce HSTS. Even though this problem has been mitigated to a large extent in this domain, the issues presented may transfer to other domains; therefore, it is critical to always remain vigilant.
- Trust Associated with Connecting to Networks: Public places may have free network hot spots, for example, but there are always risks associated with their use.
- Environment and Browser Warnings: Modern browsers will warn users if a web server is not hosting secure connections.
- Tool Familiarity: Participants will use several publicly available analysis and testing tools as well as Linux and its terminal and network configuration settings.

4. Exercise

The following exercise is presented to participants in a step-by-step fashion. Participants complete several tasks to set up the MITM and then use SSLStrip.[1] All data and systems in the exercise are fictional and simulated. Artifacts are generated using VMs that are not associated with real systems.

The following mission briefing is provided to participants.

"You are tasked with intercepting passwords or other sensitive information from unsuspecting coffee goers. Since you have a public hotspot named coffee shop people are connecting to your untrusted network."

4.1 Activity 1: View Web Browser Network Traffic

You must use Wireshark and a browser to see the network traffic between the web server and the client.

****

Kali Linux is a Debian-based Linux distribution loaded with numerous cybersecurity tools.

****

1) Start the Kali Linux machine. The username and password are *kali*.
2) Open a terminal by clicking on the terminal icon on the taskbar near the dragon (Fig. 1).

Fig. 1 Terminal location in Kali Linux

3) Start Wireshark on the Kali Linux machine by running the following command (Fig. 2):

sudo wireshark

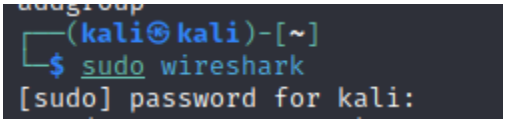

Fig. 2 Wireshark command

4) In Wireshark double-click the interface named **eth0** to start viewing the network traffic being sent and received from your machine (Fig. 3).

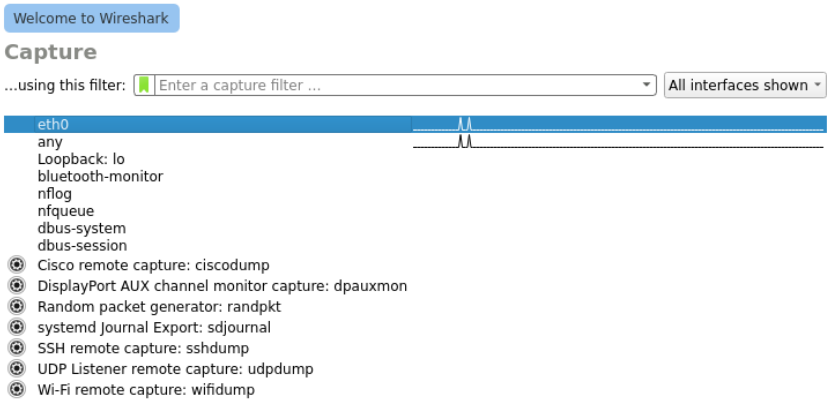

Fig. 3 Selecting eth0 interface

5) Add a filter so that only the traffic between your machine and the web server is shown. Run the following command and your output should resemble Fig. 4:

tls

Press Enter

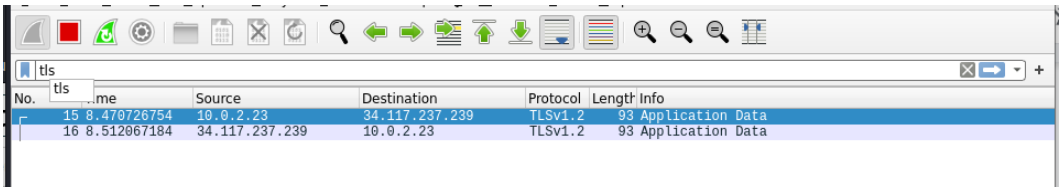

Fig. 4 Add a filter "tls" in Wireshark

6) Look for the initial handshake that your machine initiated to send data using a secure encrypted channel (Fig. 5):

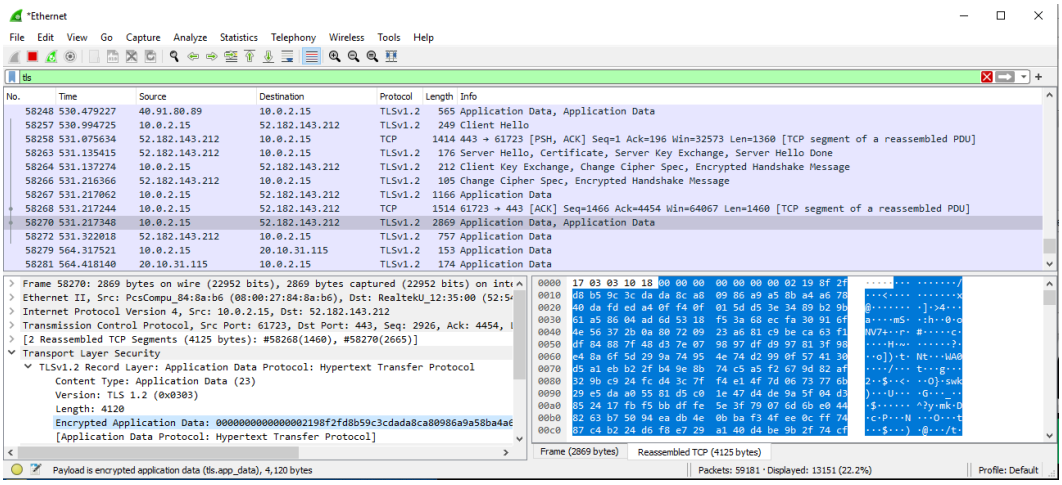

Fig. 5 Handshake between the client and web server

This activity introduced you to HTTPS and its associated traffic. Next, you will examine the difference between HTTPS and HTTP.

4.2 Activity 2: HTTP Versus HTTPS

Use Wireshark and a browser to view the network traffic and the difference between HTTP and HTTPS at the packet level.

****

HTTP and HTTPS are protocols used to fetch the resources between a web browser and web server. HTTPS is the secure version of HTTP.

****

7) Within your same Wireshark instance, in the toolbar, press the restart button and select to continue without saving. This will begin sniffing the network again. These steps are shown in Figs. 6 and 7.

Fig. 6 Restart traffic sniffing button

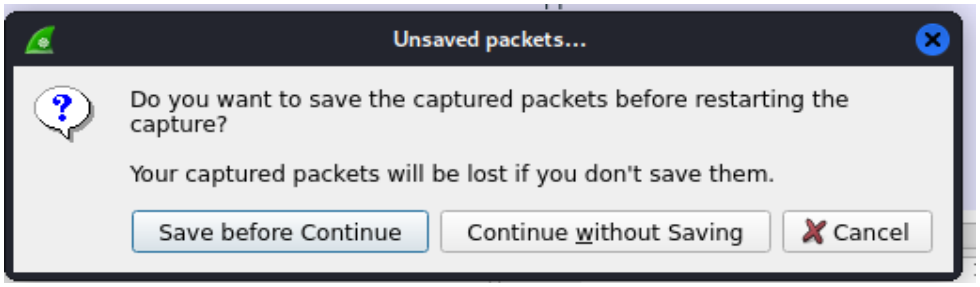

Fig. 7 After clicking restart, select continue without saving

8) Go back to your browser and navigate to the URL: http://http-password.badssl.com/. A web page should appear as shown in Fig. 8:

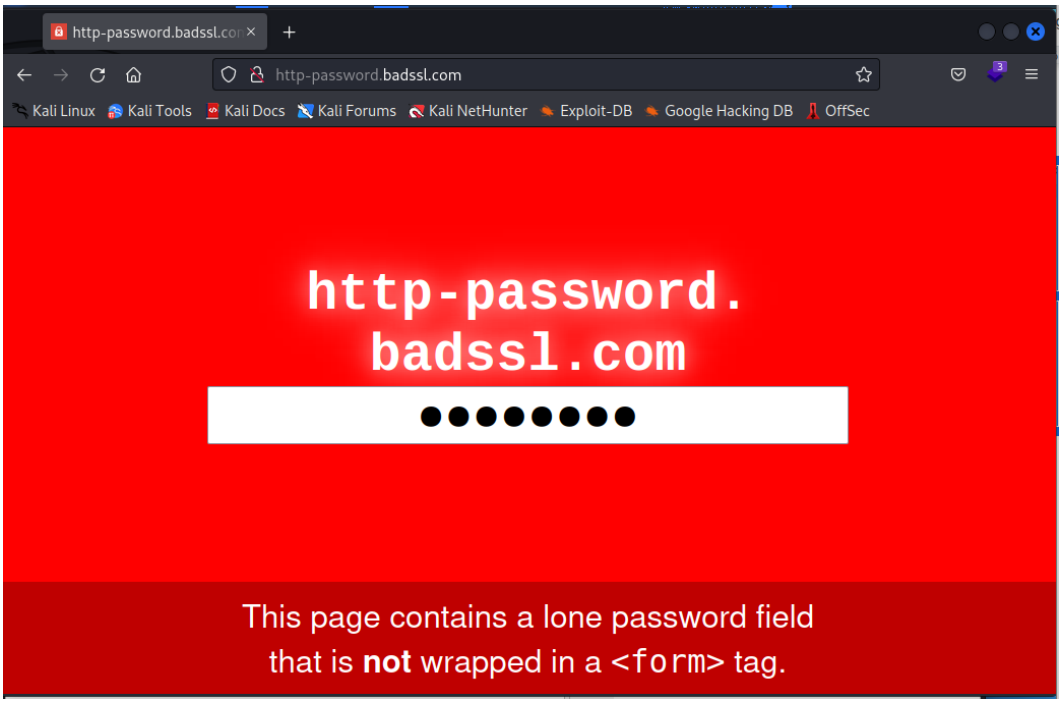

Fig. 8 The badssl website

9) The traffic generated from the request will be displayed in Wireshark. Notice the difference in the way the data is presented in comparison to HTTPS (Fig. 9):

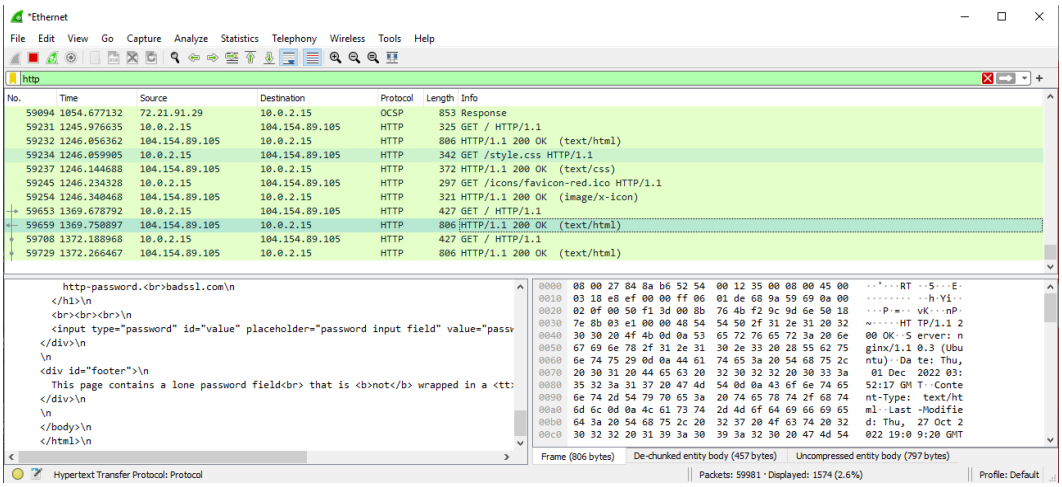

Fig. 9 HTTP traffic in Wireshark

4.3 Activity 3: Setting Up MITM Against a Windows Machine

Use arpspoof, iptables, and arp-scan to set up the Kali machine as a proxy between the client and web server.

****

All the tools used here are open-source and pre-installed on the Kali Linux Operating System.

****

10) First look for a candidate device on the network using the following command (Fig. 10).

sudo arp-scan -l –interface=eth0

Fig. 10 Command to get hosts on network

11) Something that resembles Fig. 11 should be seen. Within the list of hosts, choose the Windows 10 machine with IP address 10.0.2.15.

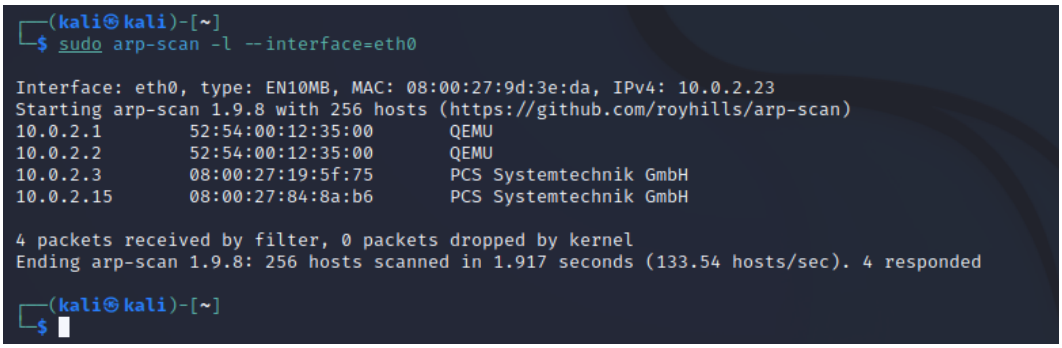

Fig. 11 Lists of hosts on the network

12) Configure your machine to enable forwarding because you will need to forward packets from the web browser to the web server and vice versa. Run the following command (Fig. 12):

sudo sysctl -w net.ipv4.ip_forward=1

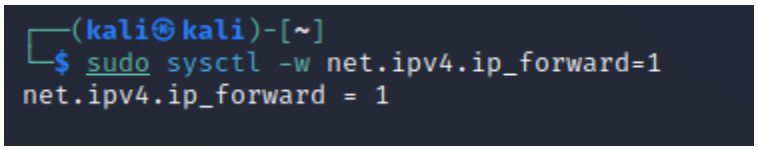

Fig. 12 Allow packet forwarding

13) Set rules using iptables that will leverage the packet forwarding just enabled. Redirect traffic from Port 80 to 8080 to act as a proxy between the client and web server by running the following command (Fig. 13):

sudo iptables -t nat -A PREROUTING -p TCP –destination-port 80 -j REDIRECT –to-port 8080

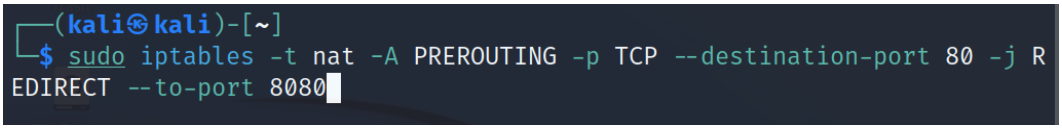

Fig. 13 Adding iptable rule to packet forward

14) Verify that the rule was added correctly by checking the routes on the machine. Run the following command (Fig. 14):

route -n

Fig. 14 Command to check routes

15) Make the Windows 10 machine route traffic through the Kali Linux machine. Your machine will pretend to be the network gateway—in other words, the traffic that leaves the Windows 10 machine will be redirected through the Kali Linux machine first. Run the following command (Fig. 15):

sudo arpspoof -i eth0 -t <IP for victim> <IP of gateway>

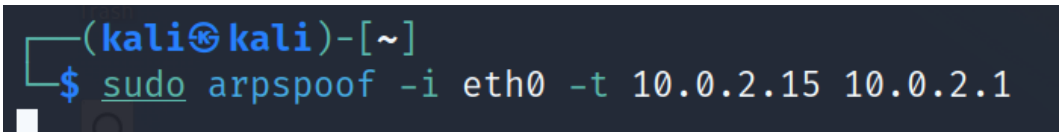

Fig. 15 arpspoof command

16) From the Windows 10 machine, run the following command to check if the spoofing is successful (Fig. 16):

arp -a

Fig. 16 Display arp table on machine

17) Now check the display of the previous command and check to see if the MAC address for the gateway and the MAC address for the machine running arpspoof are the same (Fig. 17).

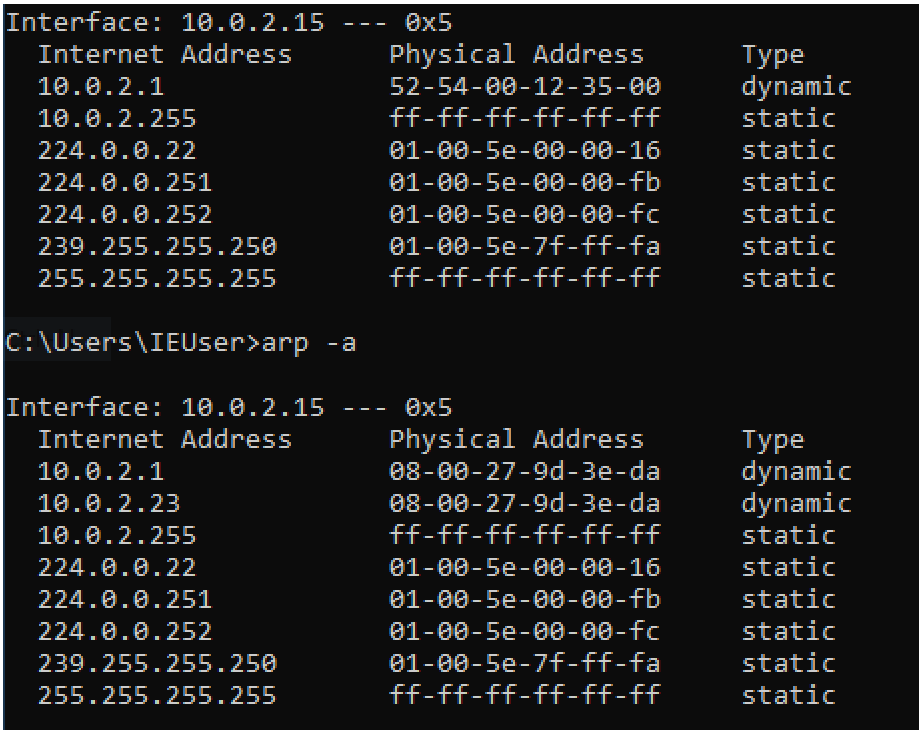

Fig. 17 arp table of machine

18) Everything is now set up for the MITM attack. Open Wireshark and enter a new filter: http. Afterward, go to the Windows 10 machine, and navigate to the URL from before: http://http-password.badssl.com/.

Back on the Kali machine, the following traffic should be seen (Fig. 18).

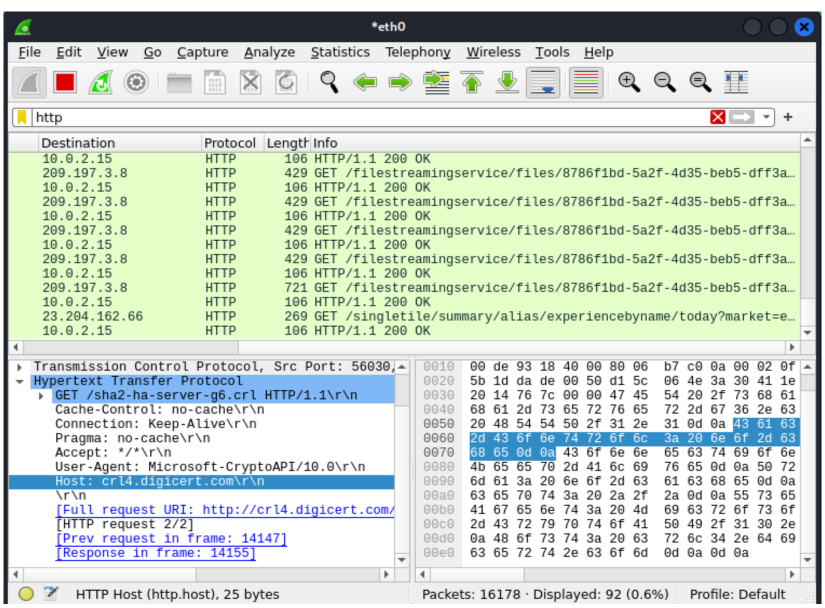

Fig. 18 Wireshark showing MITM attack

4.4 Activity 4: Perform SSL Stripping Attack

Now that the traffic is rerouted through the Kali Linux machine, the SSL stripping can be performed.

19) Recall that traffic was redirected to Port 8080; therefore, you will strip the HTTPS traffic at that port to downgrade the traffic to HTTP (Fig. 19). Ensure that the Windows 10 machine has loaded the webpage from earlier.

sslstrip -l 8080

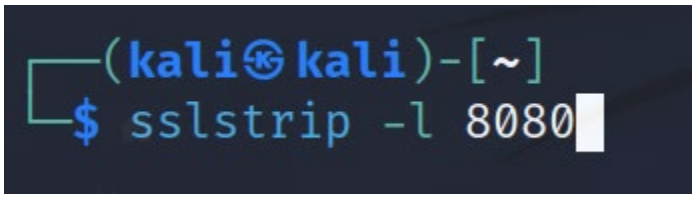

Fig. 19 sslstrip command to downgrade HTTPS traffic

20) The output of SSLStrip can be viewed interactively using the tail command. But when this is done, there will be errors because the attack will not work (Fig. 20).

tail -f -n 0 sslstrip.log

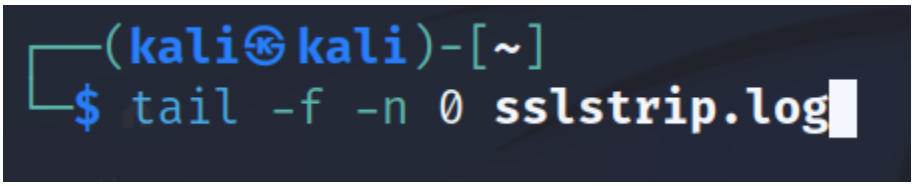

Fig. 20 View of real time logs of the attack

4.5 Activity 5: Successful SSL Stripping

This attack rarely works nowadays since most browsers have HSTS strict enforcement of HTTPS connections and request that the webpage only load over HTTPS. This disables redirects from HTTPS to HTTP, which are needed for the attack. A successful attack would have looked like the following.

21) Now that the attack was attempted on Windows 10 and failed, we will visit an intentionally vulnerable site to see the output on the Kali Linux machine (Fig. 21).

http://testphp.vulnweb.com/login.php

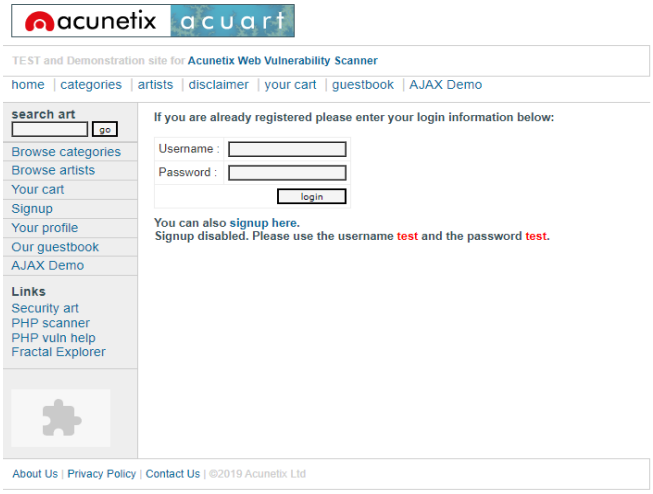

Fig. 21 Vulnerable HTTP login page to test

22) Open Wireshark and filter for HTTP once again. In the login portion, enter some random values and go back to Wireshark to see the traffic. This is where you can find your login sent in plain text.

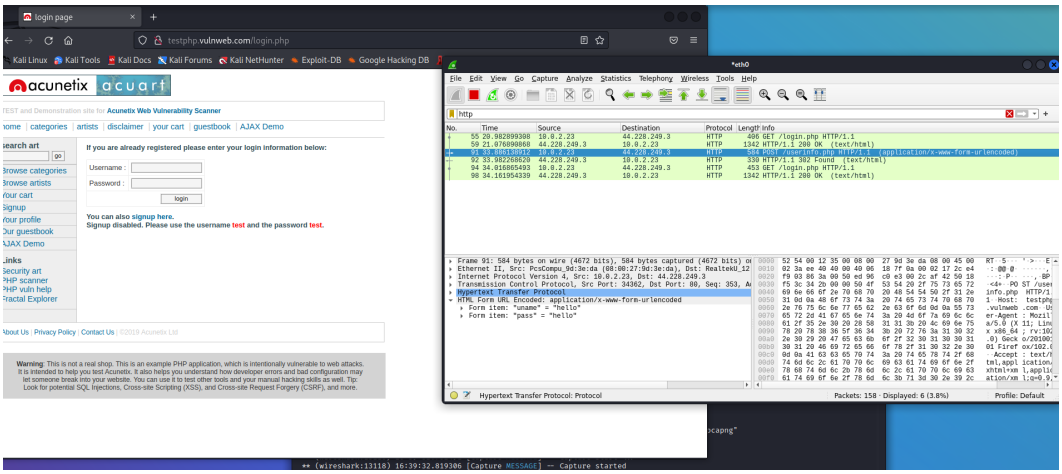

Fig. 22 Wireshark traffic of vulnerable login page

23) Think of some other network services where a MITM attack could have a similar impact as with HTTP.

_____

_____

_____

_____

24) Think of ways that you could mitigate this kind of attack in other domains.

_____

_____

Congratulations! You have successfully completed the exercise.

5. Conclusion

In this report, we have provided a basic learning module that introduces participants to web browser protocols, SSL stripping attacks, and network tools. The participant sees how these protocols are used by their browser and web server, the security of the protocols, and an example of an attack taking advantage of these protocols. By providing a hands-on exercise, participants get a firsthand look at configuring network traffic through their machine on a novel level as well as using network tools to understand the network and how the SSL stripping attack would be used.

This exercise is used for training and awareness, but also to fuel research in cybersecurity defense focused on network tools, MITM, and adaptive detection techniques.

6. References

1. Marlinspike M. New tricks for defeating SSL in practice. 2009 July [accessed 2023 Jan]. https://www.blackhat.com/presentations/bh-dc-09/Marlinspike/BlackHat-DC-09-Marlinspike-Defeating-SSL.pdf.
2. Hypertext Transfer Protocol – HTTP/1.1. 1997 Jan [accessed 2023 Jan]. https://www.rfc-editor.org/rfc/rfc2068.
3. HTTP Over TLS. 2000 May [accessed 2023 Jan]. https://www.rfc-editor.org/rfc/rfc2818.
4. HTTP Strict Transport Security (HSTS). 2012 Nov [accessed 2023 Jan]. https://www.rfc-editor.org/rfc/rfc6797.
5. The Secure Sockets Layer (SSL) Protocol Version 3.0. 2011 Aug [accessed 2023 Jan]. https://www.rfc-editor.org/rfc/rfc6101.
6. The Transport Layer Security (TLS) Protocol Version 1.3. 2018 Aug [accessed 2023 Jan]. https://www.rfc-editor.org/rfc/rfc8446.
7. Arp-Scan. 2022 Nov [accessed 2023 Jan]. https://www.kali.org/tools/arp-scan/.
8. Dsniff. 2022 Nov [accessed 2023 Jan]. https://www.kali.org/tools/dsniff/#arpspoof.
9. iptables. 2020 Apr [access 2023 Jan]. https://help.ubuntu.com/community/IptablesHowTo?action=show&redirect=Iptables.
10. Wireshark. 2022 Nov [accessed 2023 Jan]. https://www.kali.org/tools/wireshark/.
11. Kali 2022.1. [accessed 2022 Apr]. https://www.kali.org/blog/kali-linux-2022-1-release/.
12. Windows 10. [accessed 2023 Jan]. https://www.microsoft.com/en-us/software-download/windows10.
13. VirtualBox. [accessed 2022 Apr]. https://www.virtualbox.org/.
14. Acosta JC, Clarke L, Medina S, Akbar M, Hossain MS, Free-Nelson F. Repeatable experimentation for cybersecurity moving target defense. In: Garcia-Alfaro J, Li S, Poovendran R, Debar H, Yung M, editors. International Conference on Security and Privacy in Communication Systems; Springer, Cham; c2021. p. 82–99.

List of Symbols, Abbreviations, and Acronyms

HSTS	HTTP Strict Transport Security
HTTP	hypertext transfer protocol
HTTPS	hypertext transfer protocol secure
IP	Internet Protocol
MITM	man-in-the-middle
SSL	secure sockets layer
URL	uniform resource locator
VM	virtual machine

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
1 (PDF)	DEVCOM ARL FCDD RLB CI TECH LIB
2 (PDF)	DEVCOM ARL FCDD RLA ND J ACOSTA J CLARKE
1 (PDF)	UTEP J AGUAYO