بِسْمِ اللهِ الرَّحْمٰنِ الرَّحِيْمِ

SECURITY PRO

PROJECT # 4

WAF & DVWA

SECURITY PRO

INTRODUCTION

Name: Arslan Ali

Qualification: BS Software Engineering

Cohort: IEC Cybersecurity Cohort -5

Instructor: Shahzaib Ali Khan

Project: Web Application Firewall Implementation

Firewall: Modsecurity

Web: DVWA

SECURITY PRO


```
o access official Ubuntu documentation, please visit:
ttp://help.ubuntu.com/
o mail.
sfadmin@metasploitable:~$ ifconfig
th0       Link encap:Ethernet  HWaddr 08:00:27:4b:b5:27
          inet addr:192.168.0.101  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe4b:b527/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:45 errors:0 dropped:0 overruns:0 frame:0
          TX packets:71 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5805 (5.6 KB)  TX bytes:7419 (7.2 KB)
          Base address:0xd020 Memory:f0200000-f0220000

o         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:92 errors:0 dropped:0 overruns:0 frame:0
          TX packets:92 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19393 (18.9 KB)  TX bytes:19393 (18.9 KB)

sfadmin@metasploitable:~$
```

DAMN VULNERABLE WEB APPLICATION (DVWA)

Username: admin
Password: password

In this project, I installed the Virtual box on my laptop and then install the mirror of metasploitable 2. The default username and password of the metasploitable 2 is "msfadmin" and find ip address of this machine and open it in Firefox

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.

You have logged in as 'admin'

SECURITY-PRO

TESTING THE DVWA

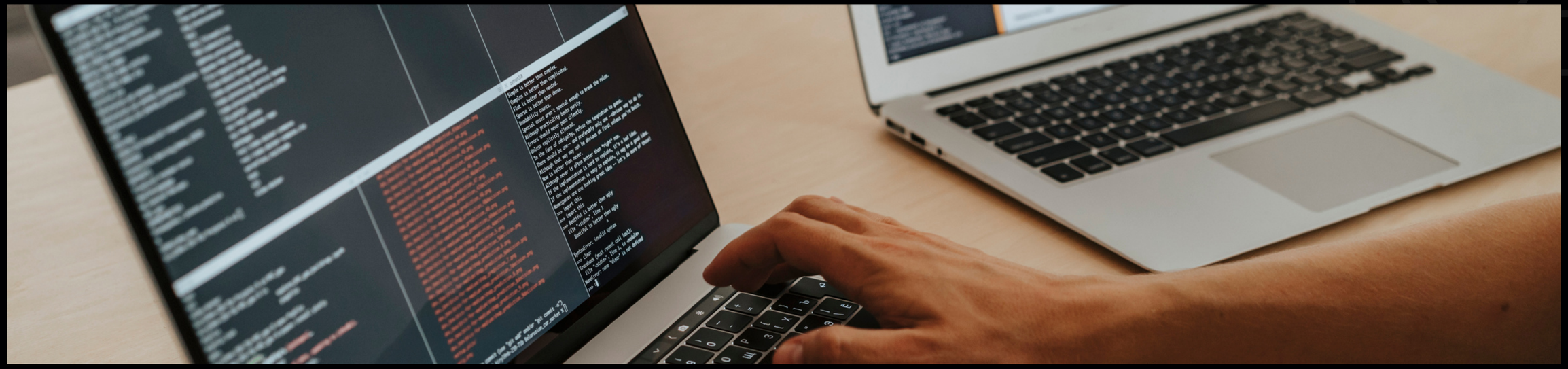
Before applying the firewall

All the attacks are performed easily like Xss, SQL injection, Command line, and Malicious file upload.

After applying the firewall

After Applying the firewall all the attacks and malicious traffic is blocked. We can see it practically

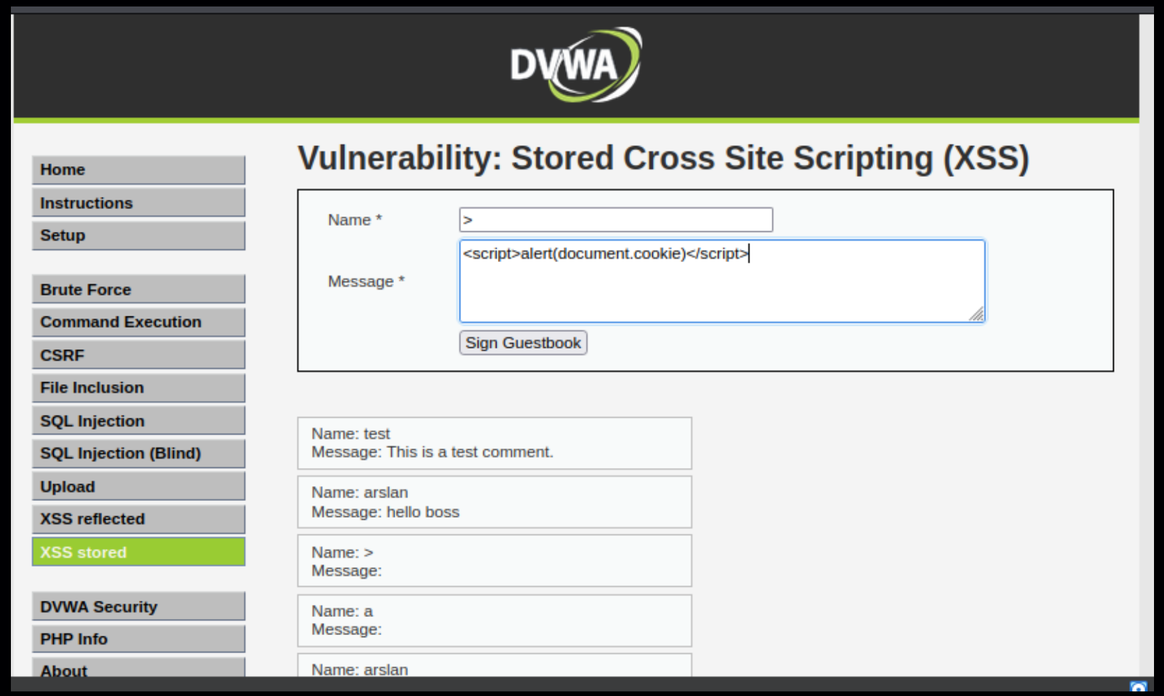
SECURITY-PRO

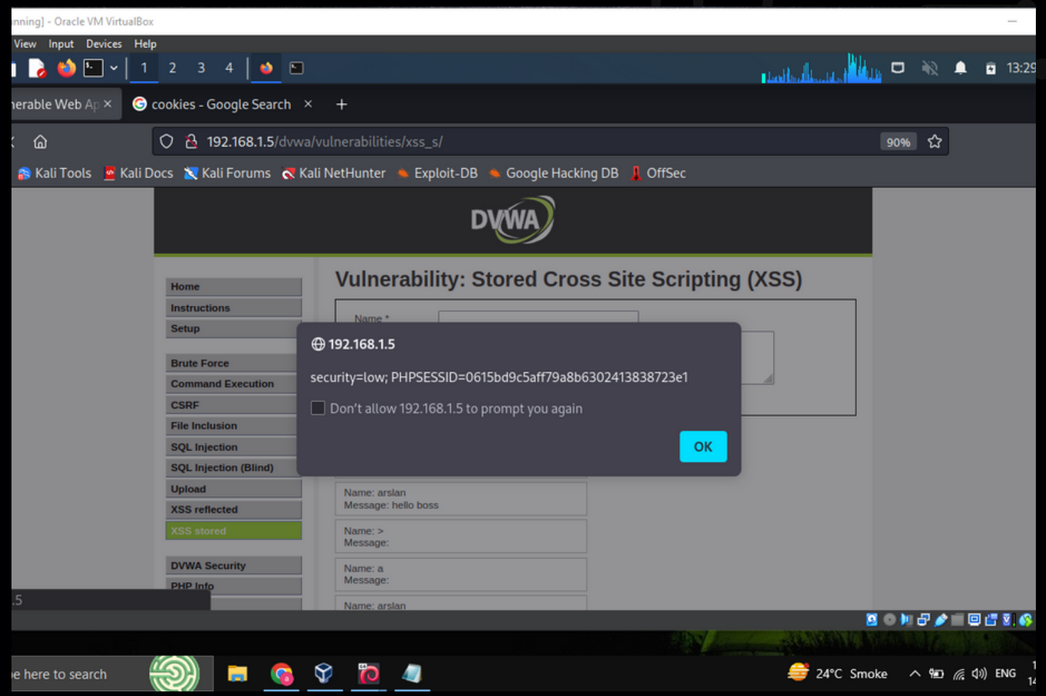


TEST DVWA WITHOUT FIREWALL

Malicious File Upload
SQL Injection & SQL Blind
XSS Reflected & Stored
HTML Injection Reflected & Stored
Command Line:
File Inclusion

XSS ATTACK(STORED)





Code:

<script>alert(document. cookie);</script>

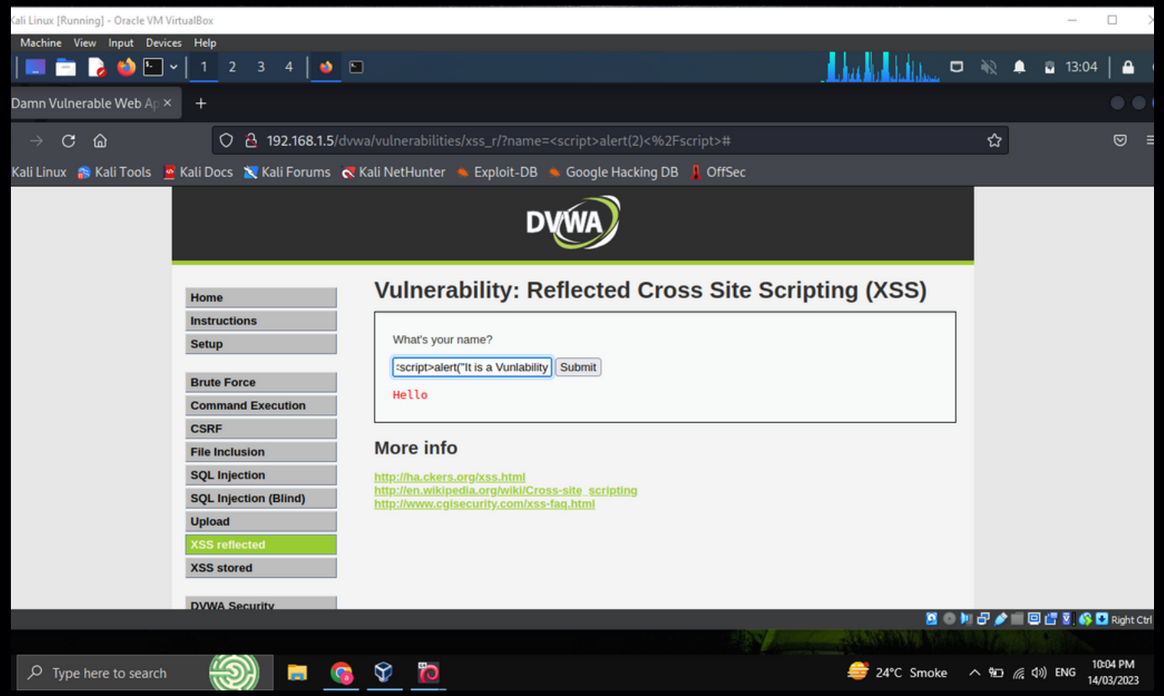
Severity:

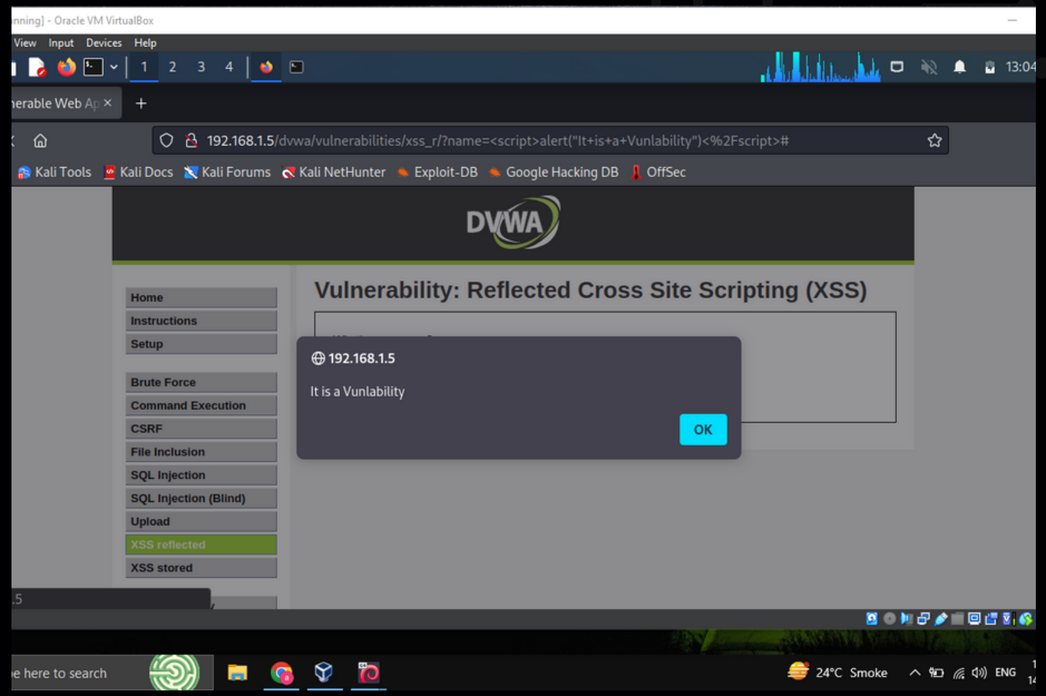
High

Risk:

- Session hijacking
- Malware distribution
- Steal cookies
- Website defacement

XSS ATTACK(REFLECTED)





Code:

```
<script>alert("Hello i am error")</script>
```

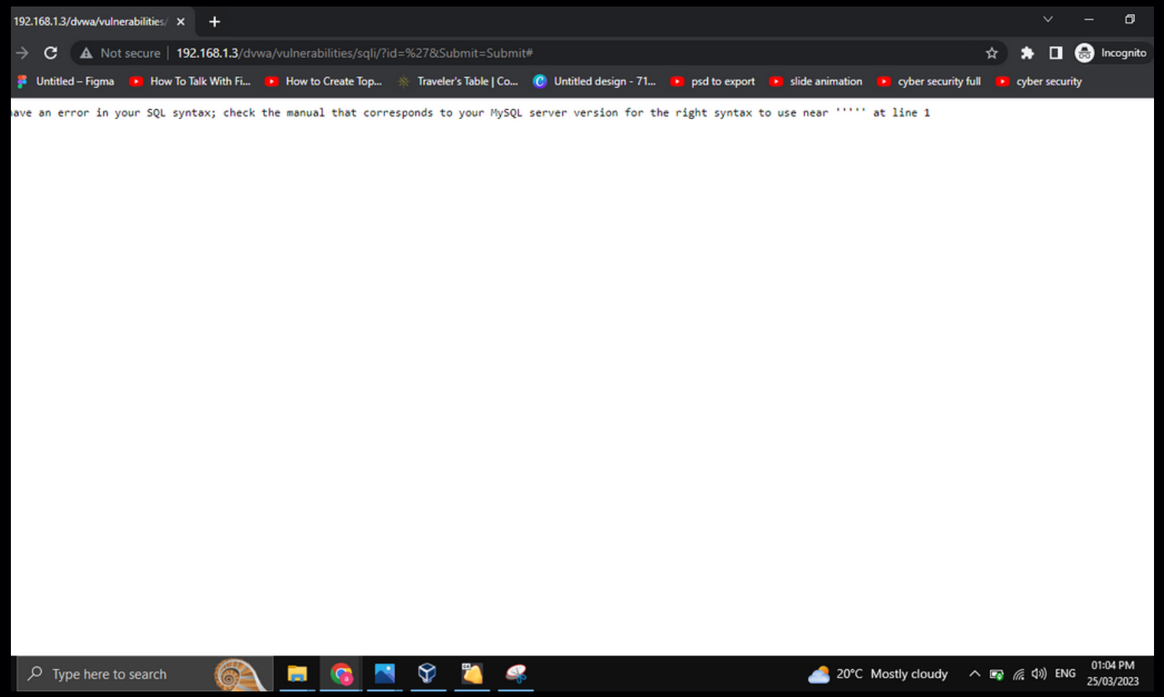
Severity:

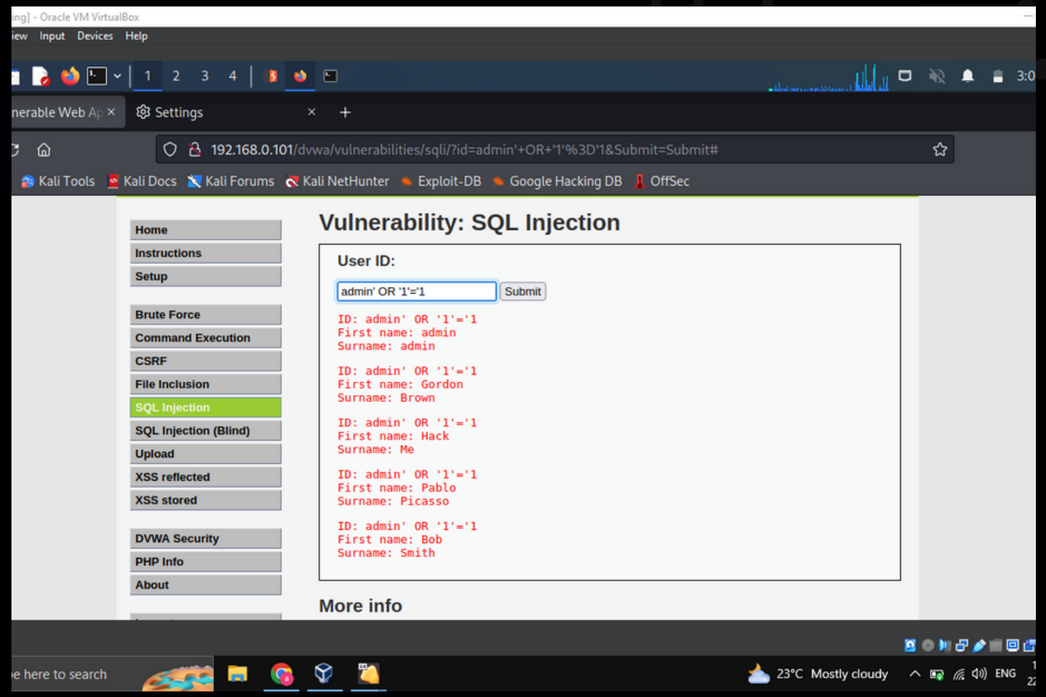
High

Risk:

- Steal sensitive information
- Steal cookies
- Modify the appearance of a website

SQL INJECTION





Code:

admin' OR '1'='1

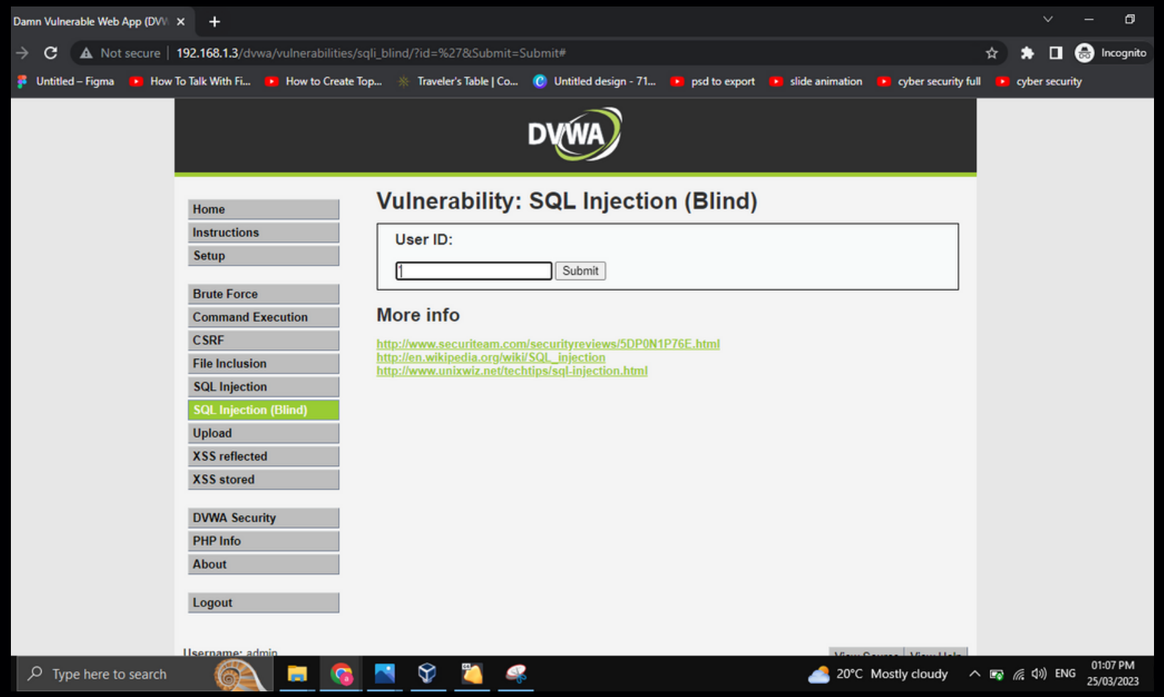
Severity:

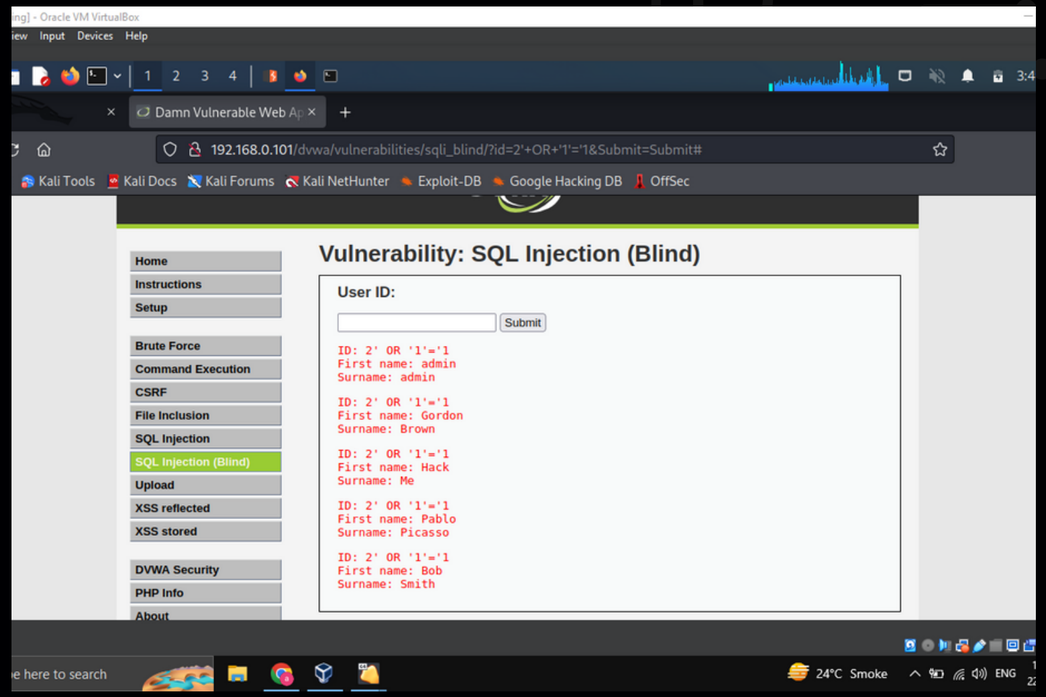
Critical

Risk:

- Unauthorized data access
- Data manipulation or deletion
- Server compromise

BLIND SQL INJECTION





Code:

admin' OR '1'='1

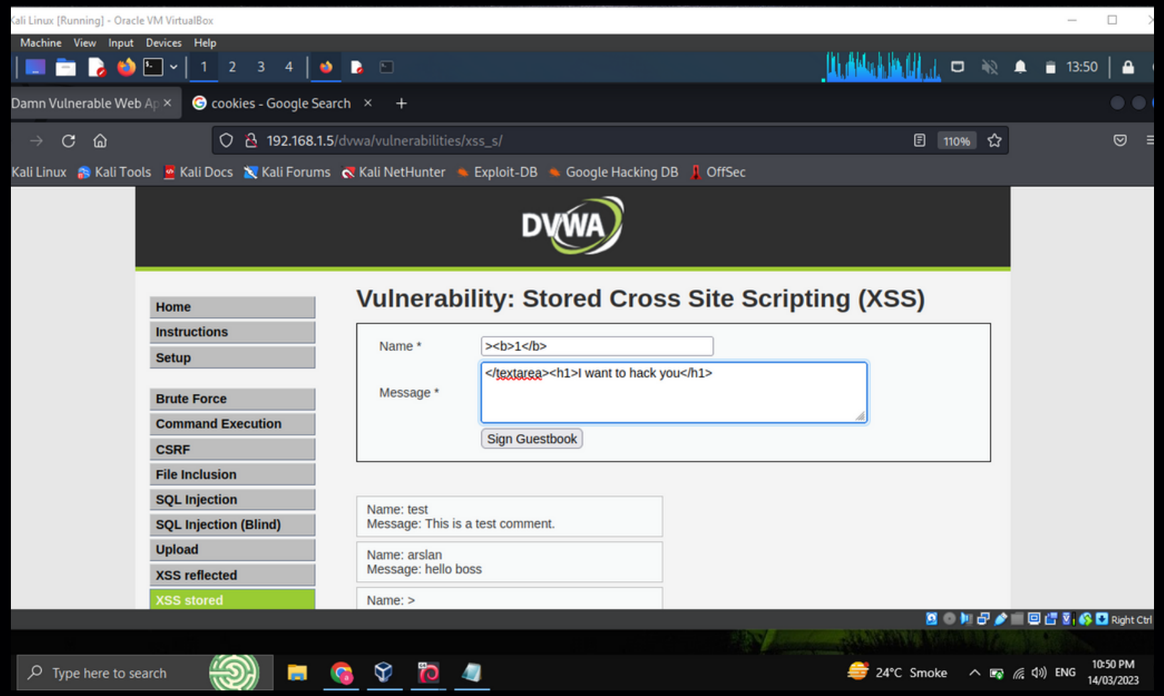
Severity:

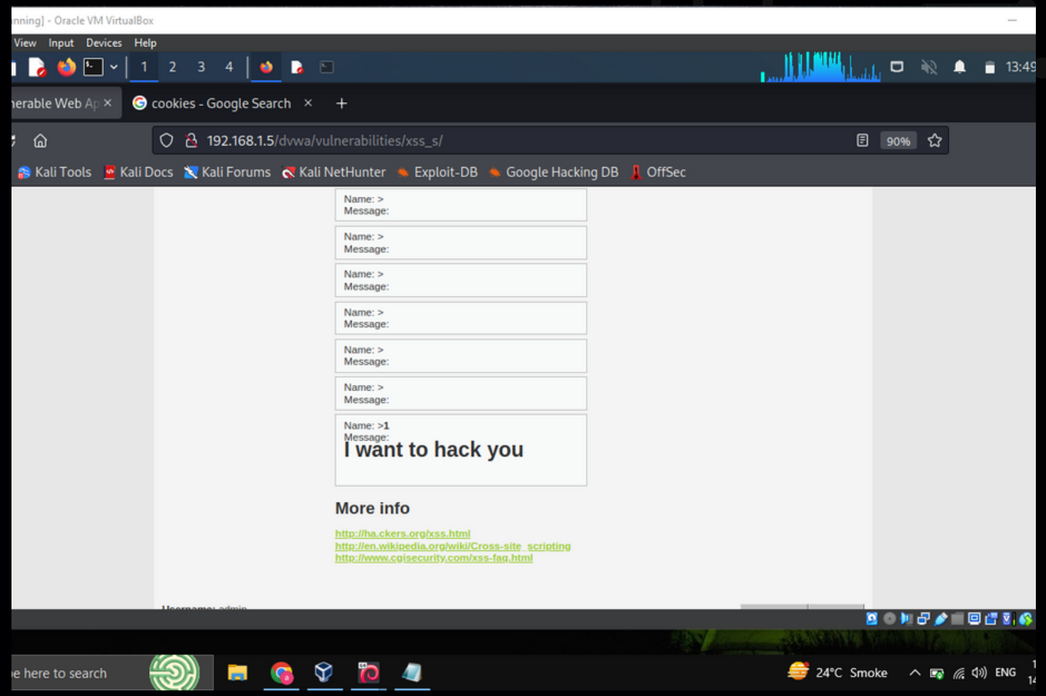
Critical

Risk:

- Unauthorized data access
- Data manipulation or deletion
- Server compromise

HTML INJECTION (STORED)





Code:

/textarea><h1>I want to hack You</h1>

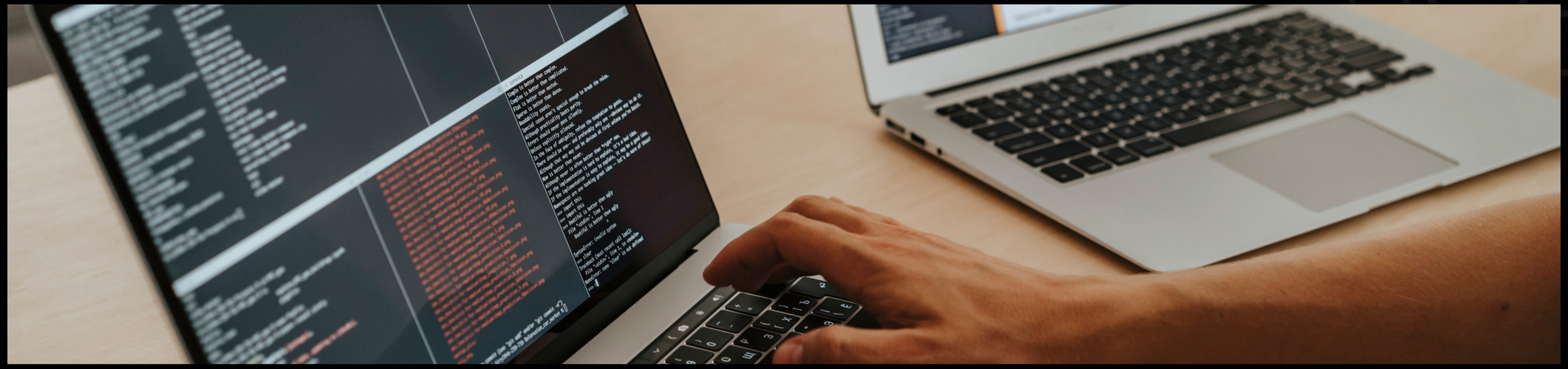
Severity:

High

Risk:

- Session hijacking
- Malware distribution
- Steal cookies
- Website defacement

SECURITY-PRO



TEST DVWA WITH FIREWALL

1st we install the ModSecurity AND after it install it and implement it and than again test the DVWA

SECURITY-PRO

INSTALL MODSECURITY

All of the 1st we update our apt in the Kali Linux terminal. After it we will install Modsecurity by using this command:

"sudo apt install libapache2-mod-security2 –y"

SECURITY-PRO

INSTALL MODSECURITY

Now we will install the the header of Apache-2 server and also restart it by using the following commands:

"sudo a2enmod headers"
"sudo systemctl restart apache2"

SECURITY-PRO

CONFIGURING MODSECURITY

Now we will Configuring ModSecurity by using the following commands:

"sudo cp /etc/modsecurity/modsecurity.conf-recommended /etc/modsecurity/modsecurity.conf
"

SECURITY-PRO

```
<VirtualHost *:80>
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        SecRuleEngine On

        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html
```

CONFIGURING MODSECURITY

We will change the file which has the following path address file and after this, we will restart the server again.

"/etc/modsecurity/modsecurity.conf"
SecRuleEngine On

SECURITY-PRO

OWASP CORE RULE SET APPLY ON FIREWALL

OWASP Core Rule Set applies on firewallModSecurity by using the following commands:

1) Remove current rules of ModSecurity:
"sudo rm -rf /usr/share/modsecurity-crs"

2) we installed the git by using this command
"sudo apt install git"

3) we Clone the OWASP-CRS GitHub repository
"sudo git clon https://github.com/coreruleset/coreruleset /usr/share/modsecurity-crs"

OWASP CORE RULE SET & ENABLING MODSECURITY

1) Now we should rename the” crs-setup.conf”.example to “crs-setup.conf” by the following command:
"sudo mv /usr/share/modsecurity-crs/crs-setup.conf.example /usr/share/modsecurity-crs/crs-setup.con"

2) Rename the default request exclusion rule file by using this command:
"sudo mv /usr/share/modsecurity-crs/rules/REQUEST-900-EXCLUSION-RULES-BEFORE-CRS.conf.example /usr/share/modsecurity-crs/rules/REQUEST-900-EXCLUSION-RULES-BEFORE-CRS.conf"

3) After this we must be edit the file "security2.conf" and paste the following code:
"“SecDataDir /var/cache/modsecurity Include /usr/share/modsecurity-crs/crs-setup.conf Include /usr/share/modsecurity-crs/rules/*.conf”

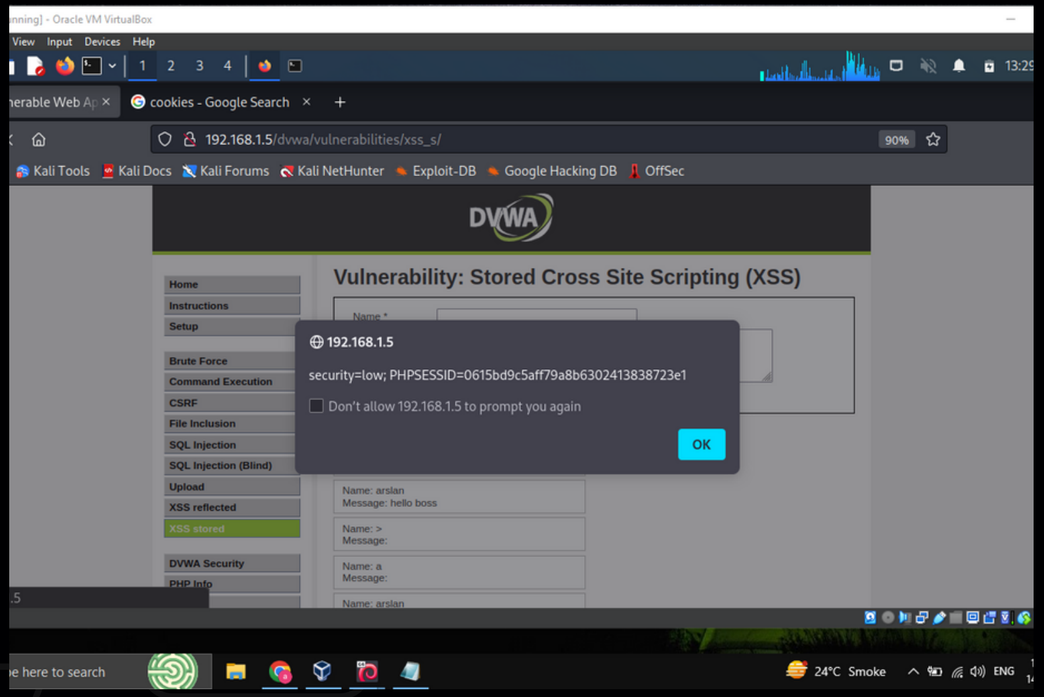
3) After this we must be edit the file "/etc/apache2/sites-enabled/000-default.conf " and paste the following code:
"“ ServerAdmin webmaster@localhost
DocumentRoot /var/www/html
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

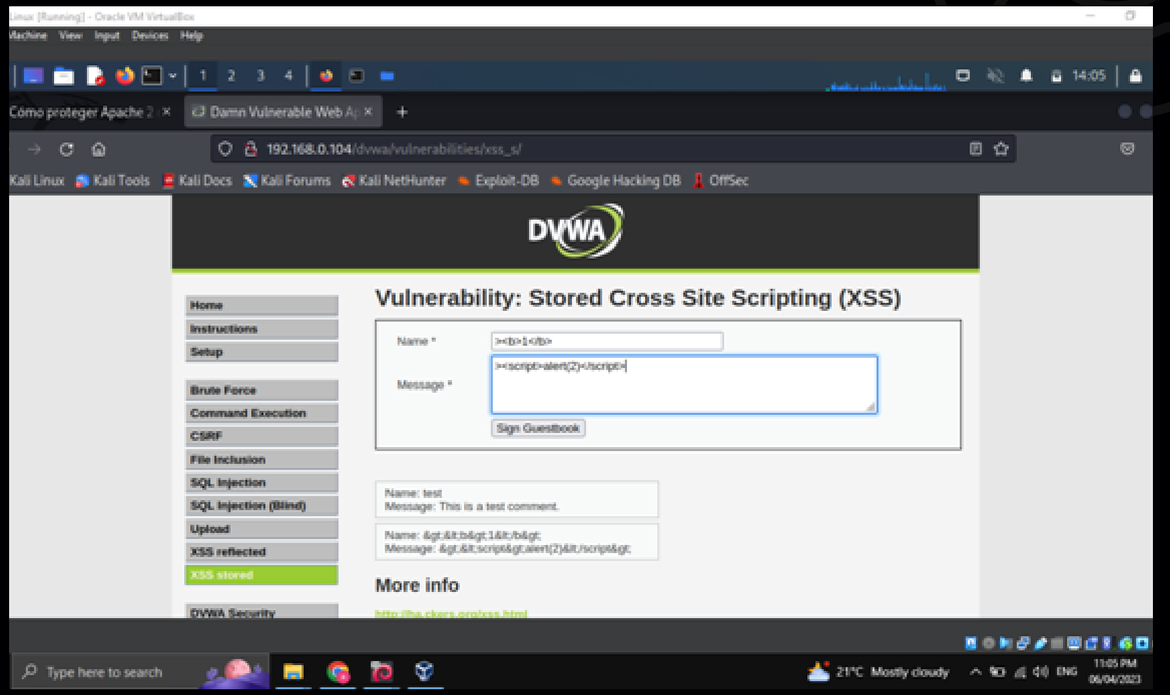
SECURITY-PRO

```
1 # -- Rule engine initialization ----------------------------------------------
2
3 # Enable ModSecurity, attaching it to every transaction. Use detection
4 # only to start with, because that minimises the chances of post-installation
5 # disruption.
6 #
7 SecRuleEngine On
8
9
10 # -- Request body handling ---------------------------------------------------
11
12 # Allow ModSecurity to access request bodies. If you don't, ModSecurity
13 # won't be able to see any POST parameters, which opens a large security
14 # hole for attackers to exploit.
15 #
16 SecRequestBodyAccess On
17
18
19 # Enable XML request body parser.
20 # Initiate XML Processor in case of xml content-type
21 #
```

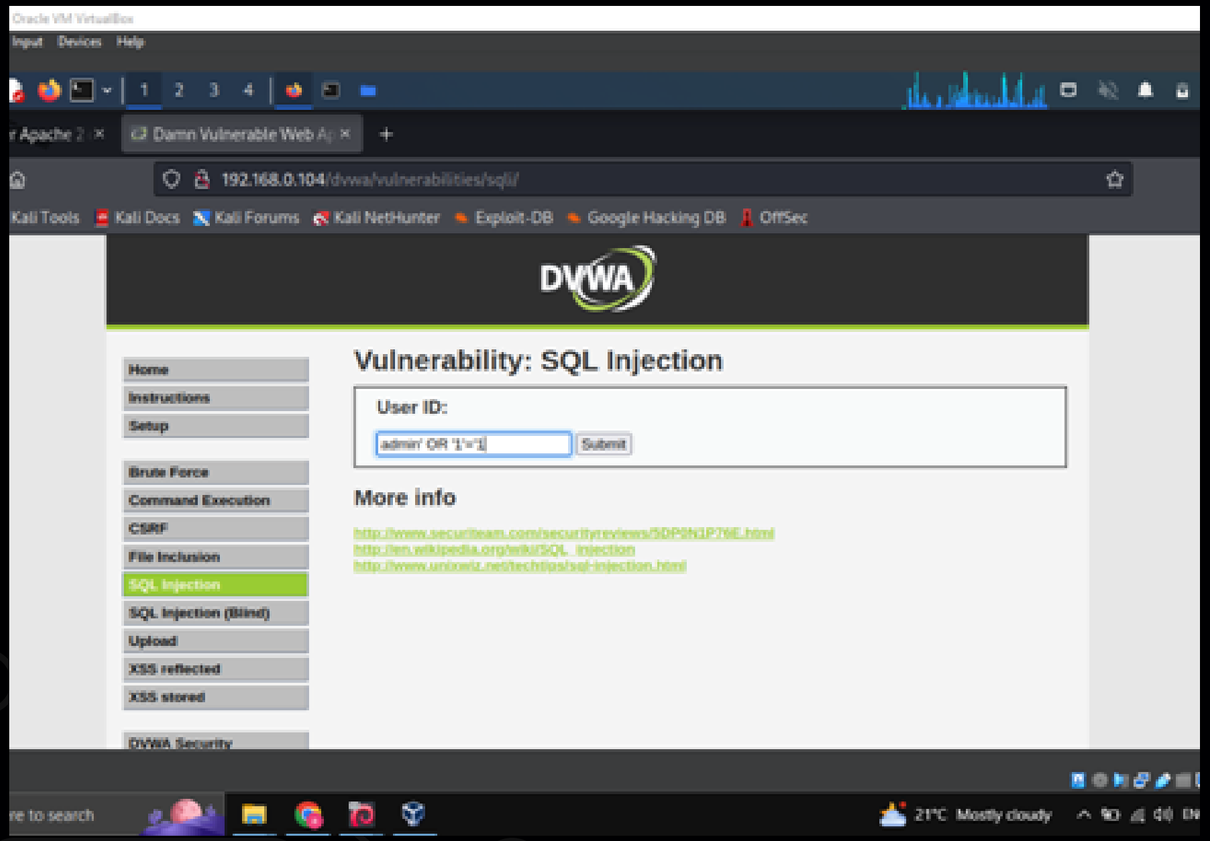
NOW ITS TIME TO TEST THE FIREWALL MODSECURITY

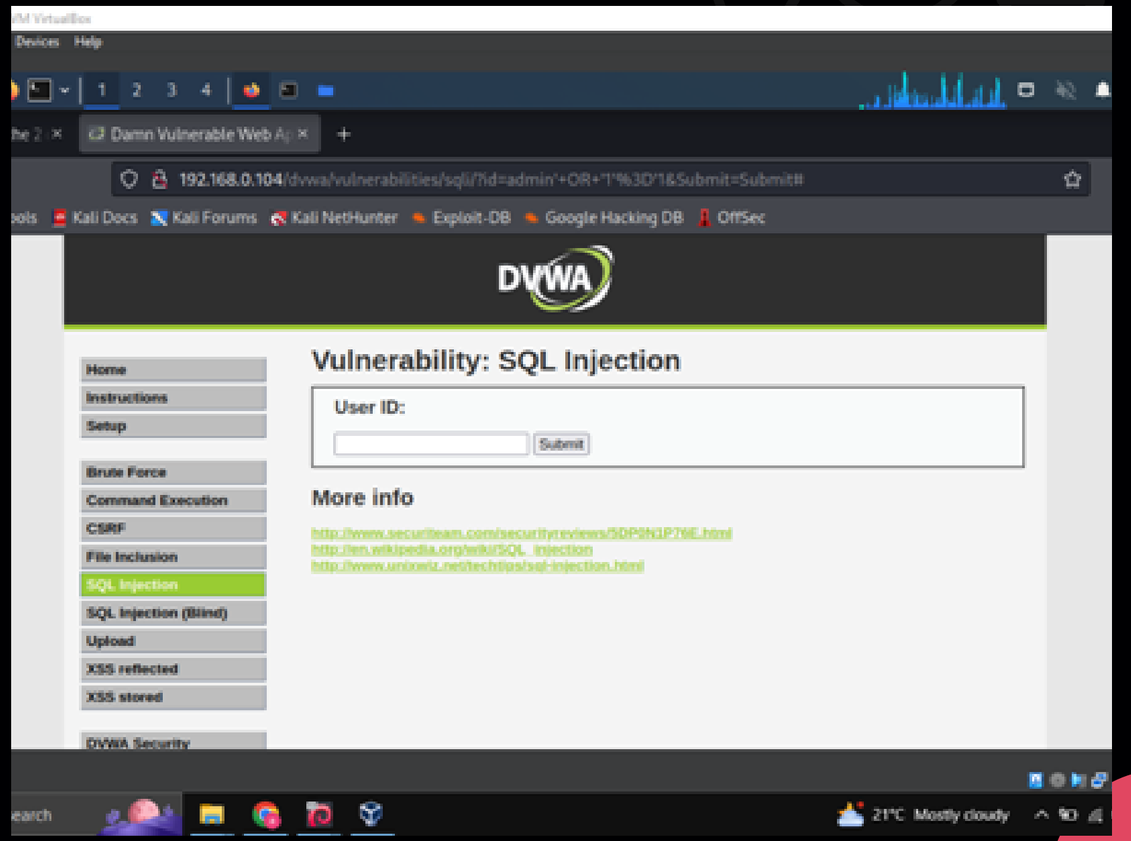
XSS ATTACK(STORED)



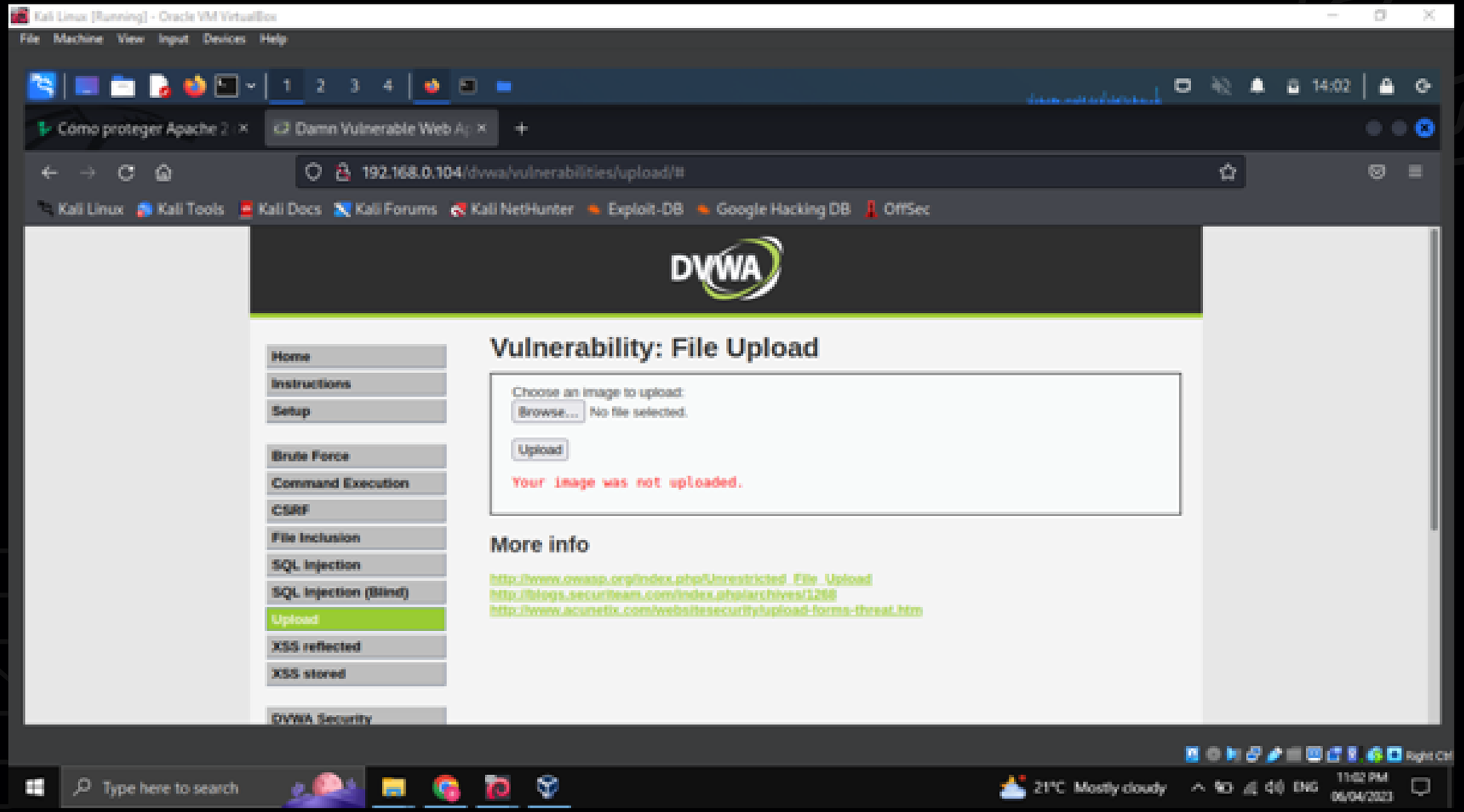


SQL INJECTION

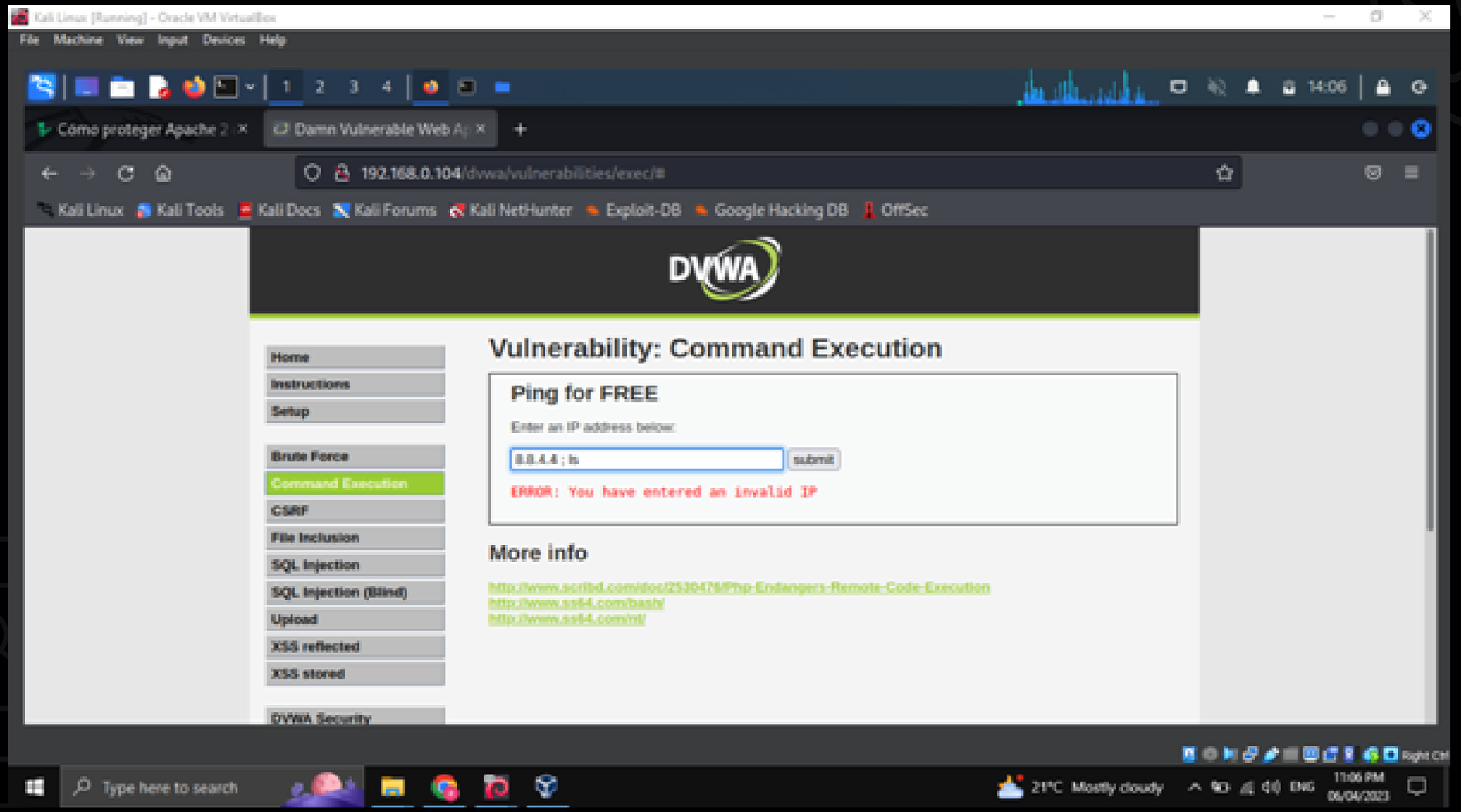




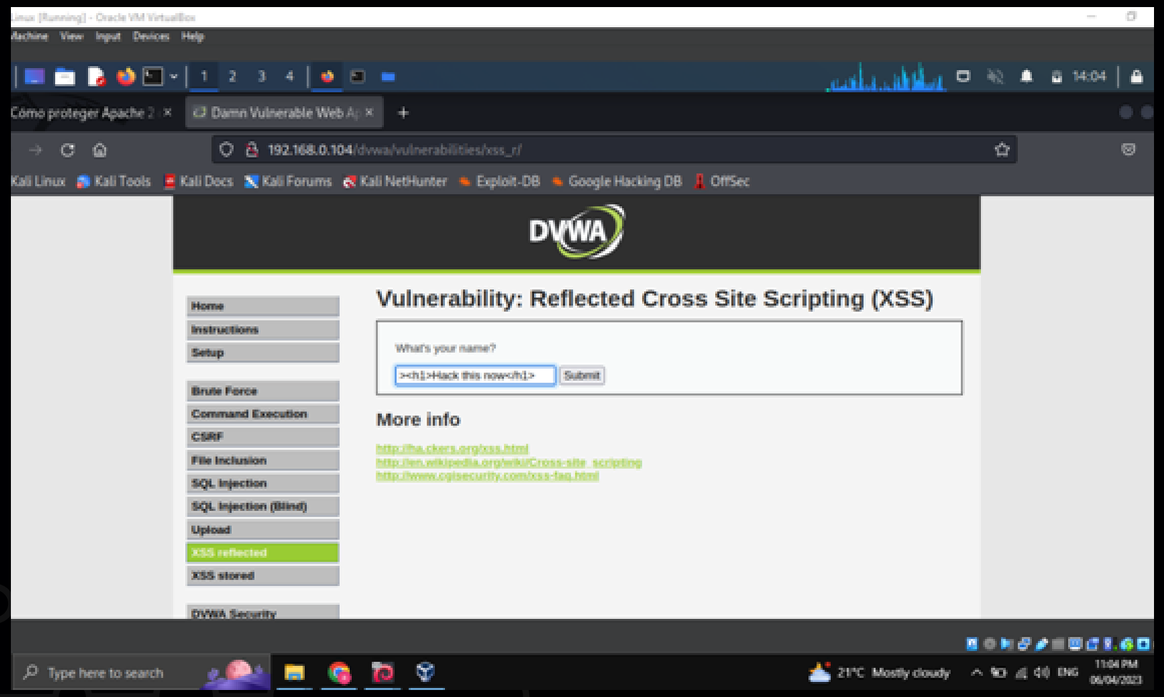
MALICIOUS FILE

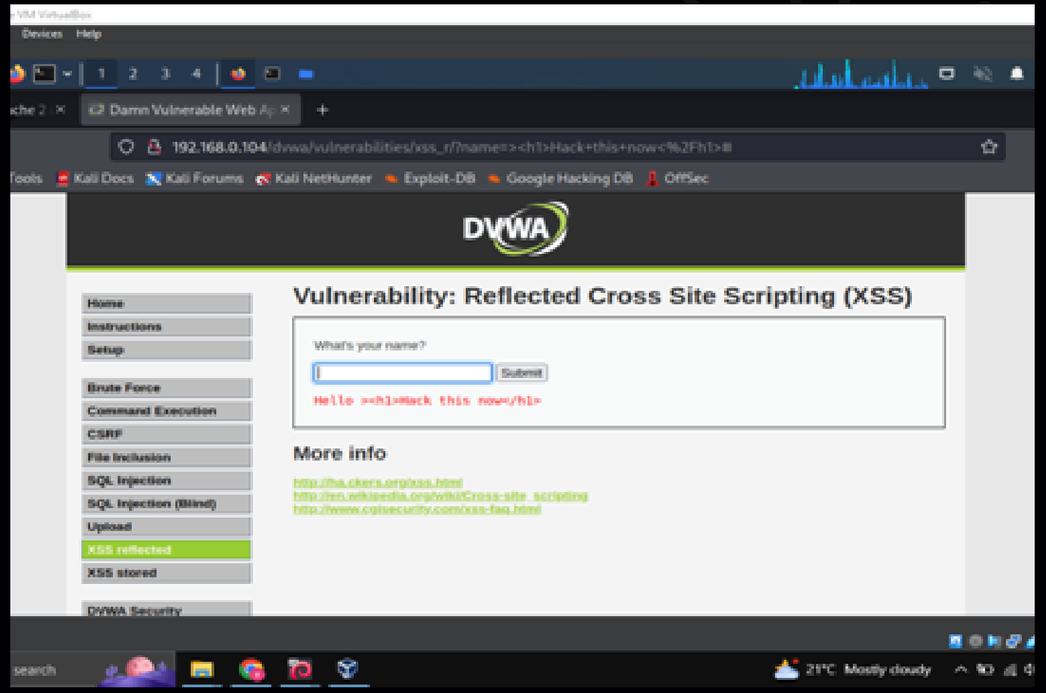


COMMAND LINE:



HTML INJECTIONS





WAF PROJECT 4

THANK YOU

SECURITY PRO