Web Application Security: Understanding Threats and Mitigating File Upload Risks

Capstone Project

Presented by Blake Major

Institute of Data

Contents

- Penetration testing overview
- Steps to a pentest
  - Reconnaissance
  - Enumeration / Scanning
  - Exploitation
  - Privilege Escalation
  - Post Exploitation / Reporting
- Regulations
- Web Application Attacks
- Common Attacks
- Exploiting a vulnerable web application
  - Creating a test environment
  - Downloading and configuring payload
  - Uploading malicious file
  - Establishing reverse shell
- Mitigation
  - In this scenario
  - For Web attacks as a whole
- Conclusion

Pentesting Overview

A penetration test (pentest) ethically tests security defenses using tools and techniques similar to those of malicious attackers, akin to an audit.

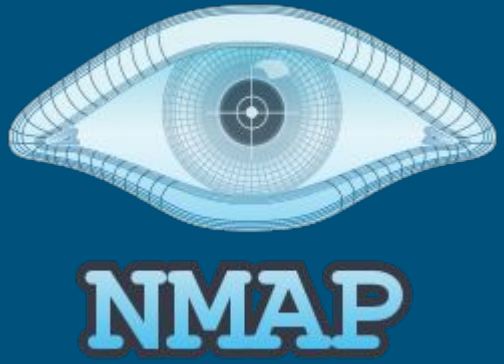
A pentest generally comprises of 5 steps:

1. Gathering information / Reconnaissance
2. Enumeration / Scanning
3. Exploitation
4. Privilege Escalation
5. Post-exploitation / Reporting

Reconnaissance

- **Information Gathering:** Collect as much data as possible about the target system, including network topology, operating systems, and applications
- **Active and Passive Methods:** Use both passive (publicly available information) and active (direct interaction with the target) techniques to gather information
- **Planning Attacks:** Utilize the gathered information to plan effective attack strategies and identify potential vulnerabilities

Enumeration / Scanning



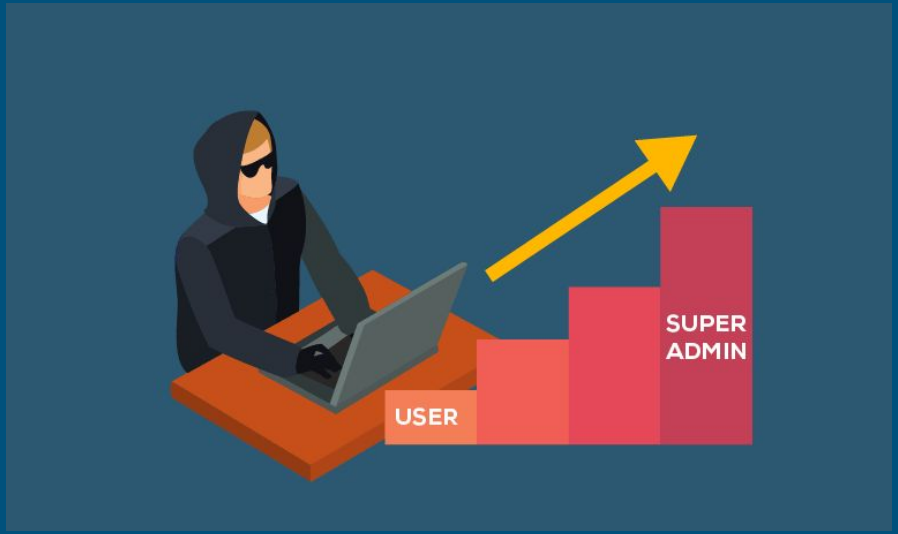
- **Identify Open Ports:** Use tools like Nmap to detect open ports and services running on the target system
- **Network Mapping:** Analyze network traffic and map the network topology to understand the target’s structure
- **Vulnerability Scanning:** Perform vulnerability scans to identify potential security weaknesses in the detected services

Exploitation

- **Exploiting Vulnerabilities:** Use identified vulnerabilities to breach the target system, often through techniques like SQL injection, cross-site scripting (XSS), or backdoors

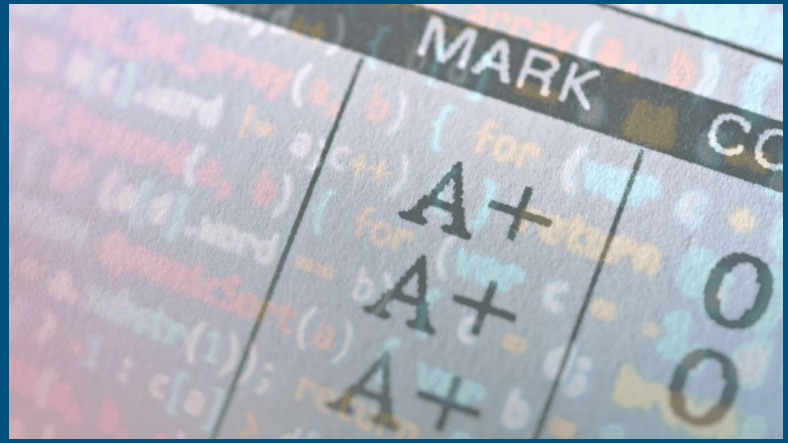
- **Data and System Access:** Access sensitive data and critical system components to understand the potential impact of a successful attack

Privilege Escalation



- **Elevate Access Rights:** Gain higher-level permissions to access more sensitive parts of the system
- **Exploit Misconfigurations:** Use weaknesses in system configurations or software to escalate privileges
- **Bypass Access Controls:** Circumvent security measures to perform actions that are normally restricted

Post Exploitation / Reporting



- **Document Findings:** Compile detailed reports on the vulnerabilities exploited, methods used, and data accessed during the test
- **Risk Analysis:** Assess the potential impact of the discovered vulnerabilities on the organization
- **Recommendations:** Provide actionable recommendations for remediation to improve the security posture

Regulations

A penetration testing engagement is guided by a series of frameworks and rules to prevent misuse or malicious actions

- Each pentest begin with the Rules of Engagement (ROE) which is an agreement between the pentester and client which outline the
  - Permission
  - Rules
  - Scope
- Frameworks are often utilised to ensure reliability and accuracy

Web Application Attacks...

- Web application attacks account for around 39% of all breaches
- Will cost the world an estimated $4 trillion in 2025
- Are extremely accessible attack targets

Common Web Application Attacks

Cross Site Scripting (XSS)

- Where a user uploads malicious code to a website that is then replicated in other users sessions
- Extremely versatile attack - can achieve anything
- Difficult to completely prevent

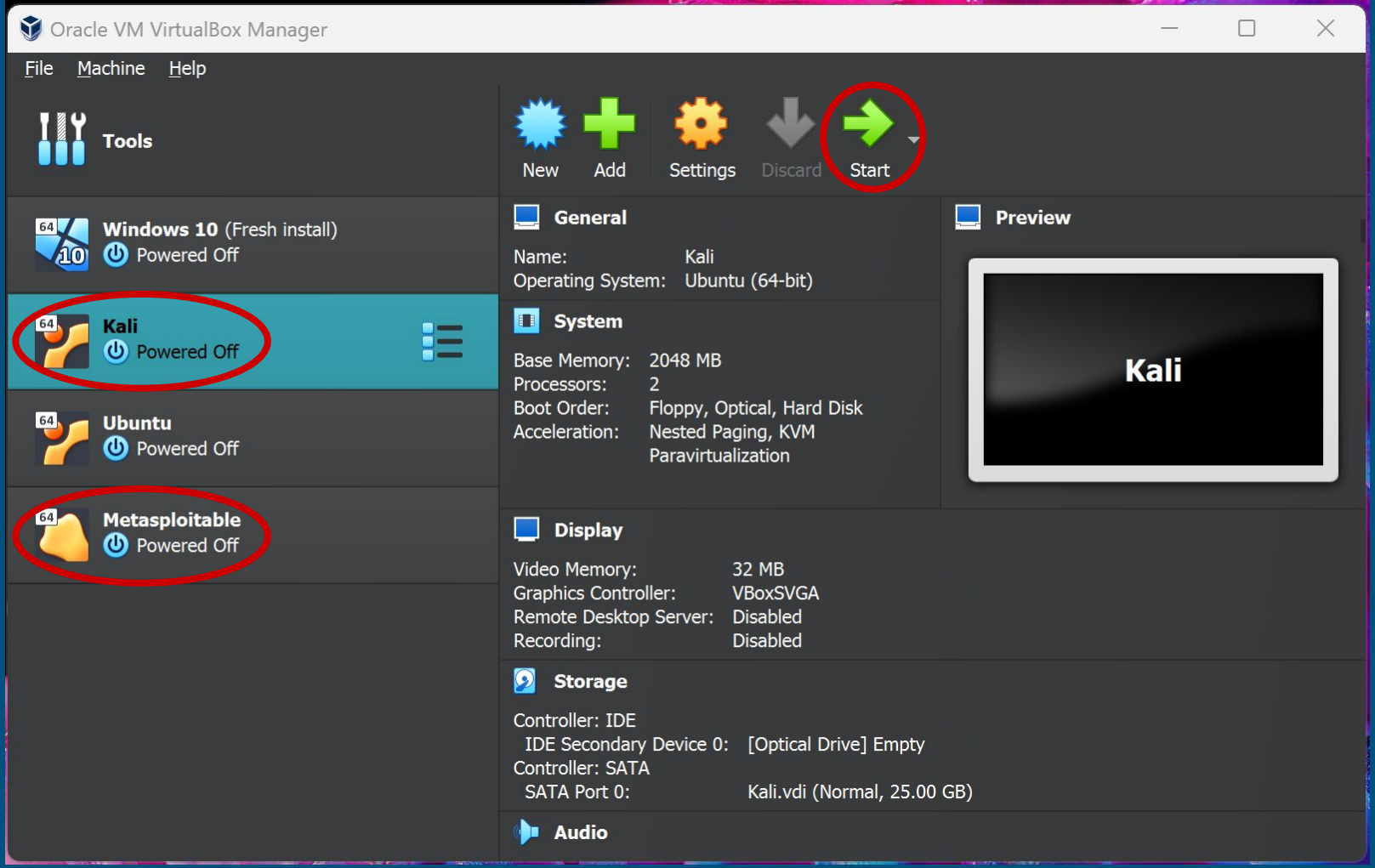
Cross Site Request Forgery (CSRF)

- Attackers tricks your browser into sending a request you did not make
- Less common in the modern era
- Still extremely effective if there are no measures to prevent it

SQL Injections

- Malicious code is “injected” into a web app and interferes with its database
- Still extremely prevalent today
- One of the most successful and most impactful attacks

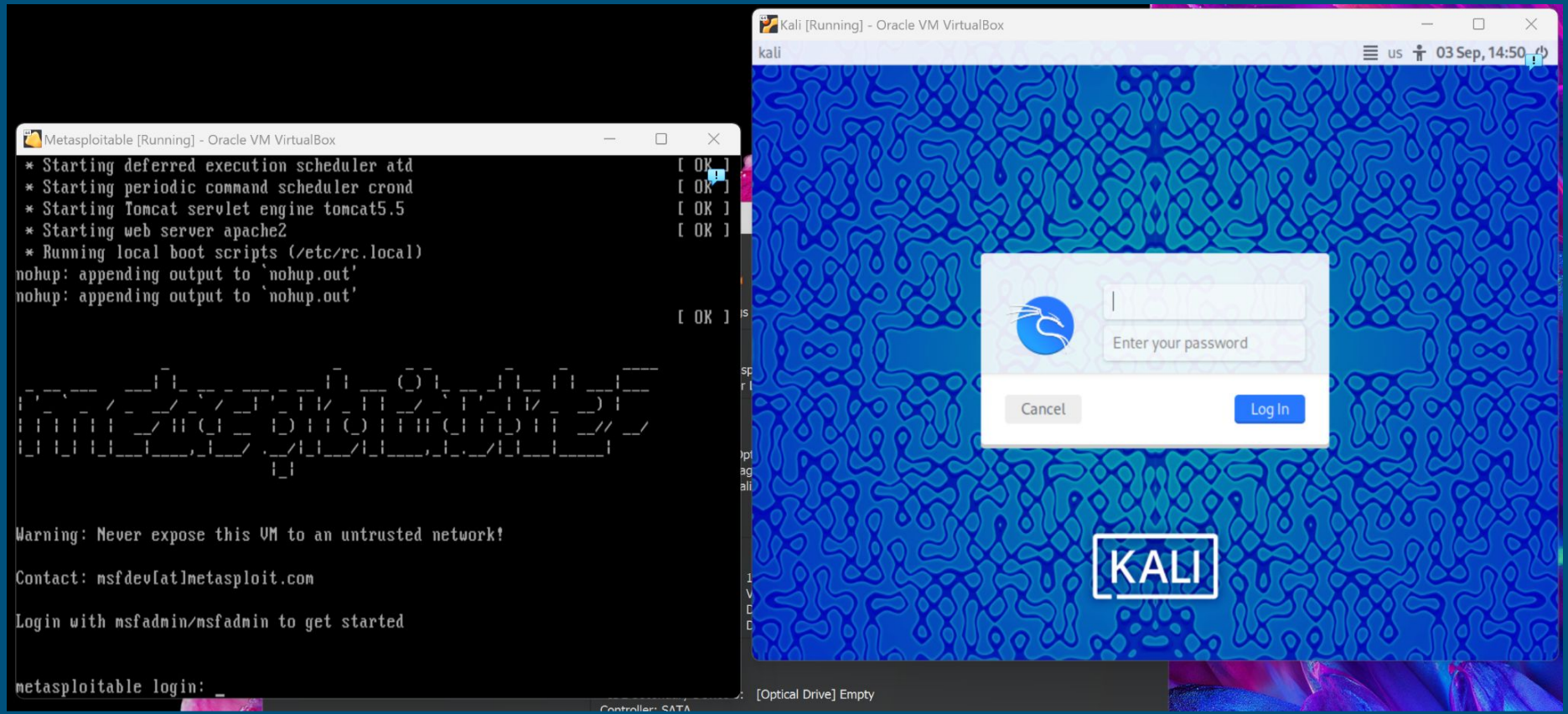
Exploiting unrestricted uploads

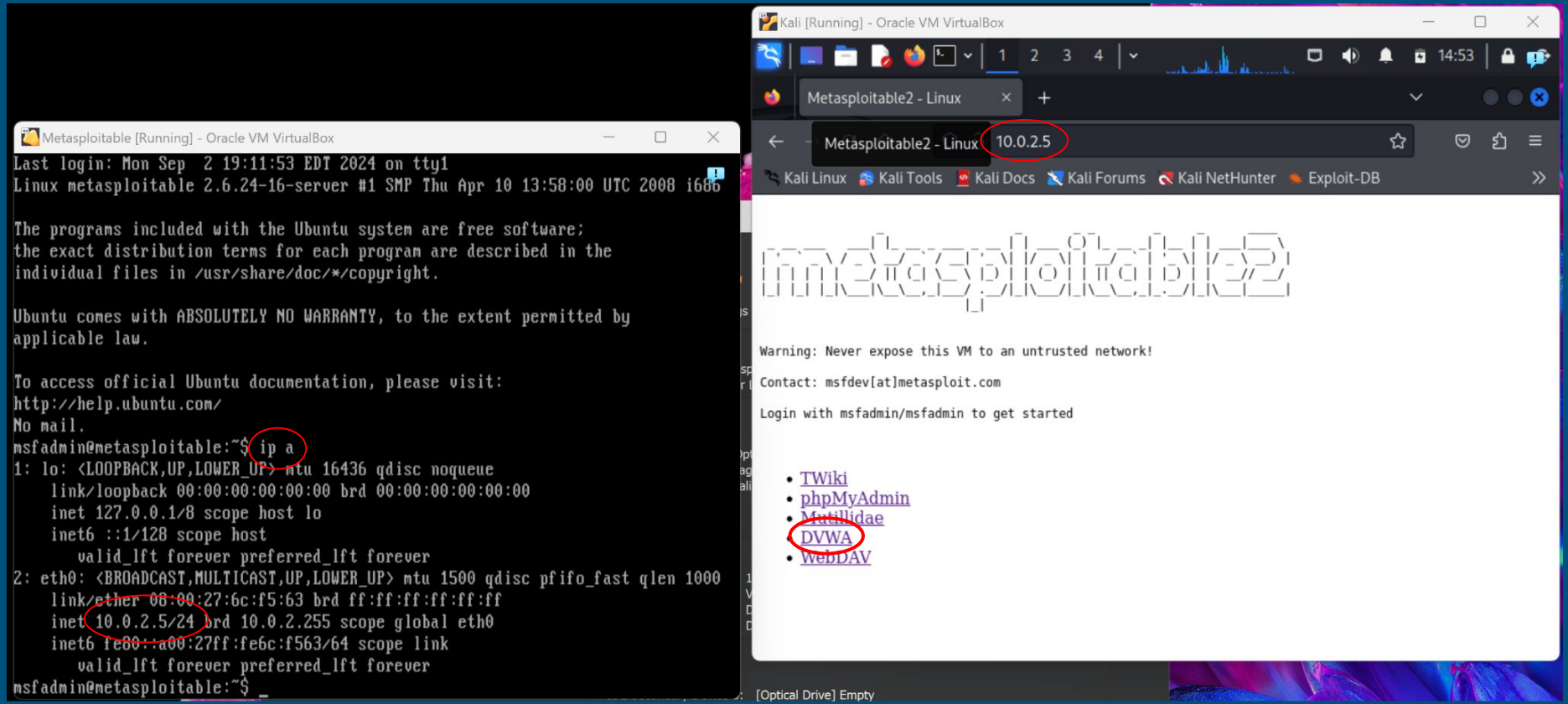

Oracle VM VirtualBox Manager
File
Machine
Help
Tools
Windows 10 (Fresh install)
Powered Off
Kali
Powered Off
Ubuntu
Powered Off
Metasploitable
Powered Off
New
Add
Settings
Discard
Start
General
Name: Kali
Operating System: Ubuntu (64-bit)
System
Base Memory: 2048 MB
Processors: 2
Boot Order: Floppy, Optical, Hard Disk
Acceleration: Nested Paging, KVM
Paravirtualization
Preview
Kali
Display
Video Memory: 32 MB
Graphics Controller: VBoxSVGA
Remote Desktop Server: Disabled
Recording: Disabled
Storage
Controller: IDE
IDE Secondary Device 0: [Optical Drive] Empty
Controller: SATA
SATA Port 0: Kali.vdi (Normal, 25.00 GB)
Audio

Metasploitable

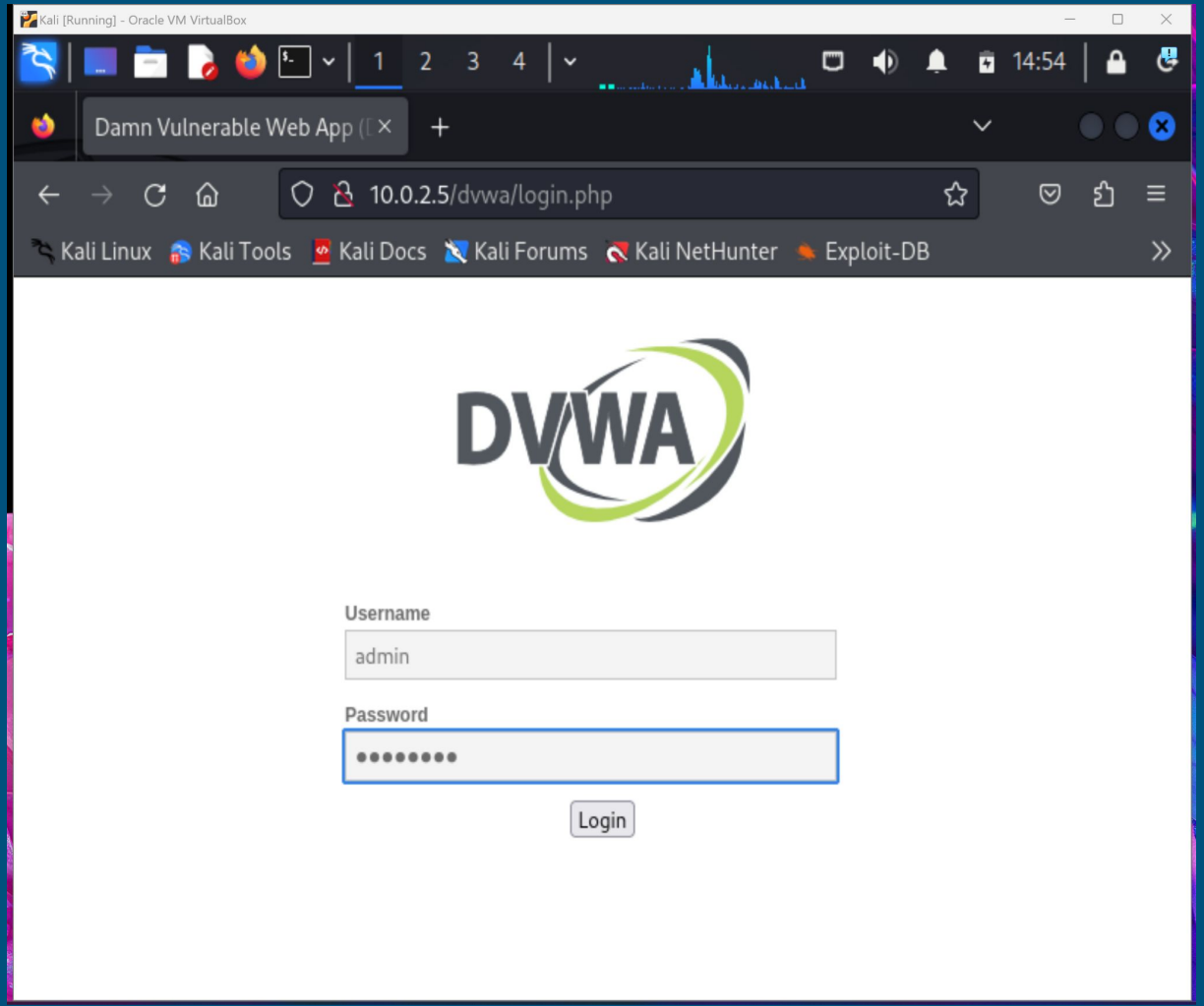
- An intentionally vulnerable Ubuntu Linux virtual machine developed by Rapid7

- It provides a secure environment to conduct security tests and research

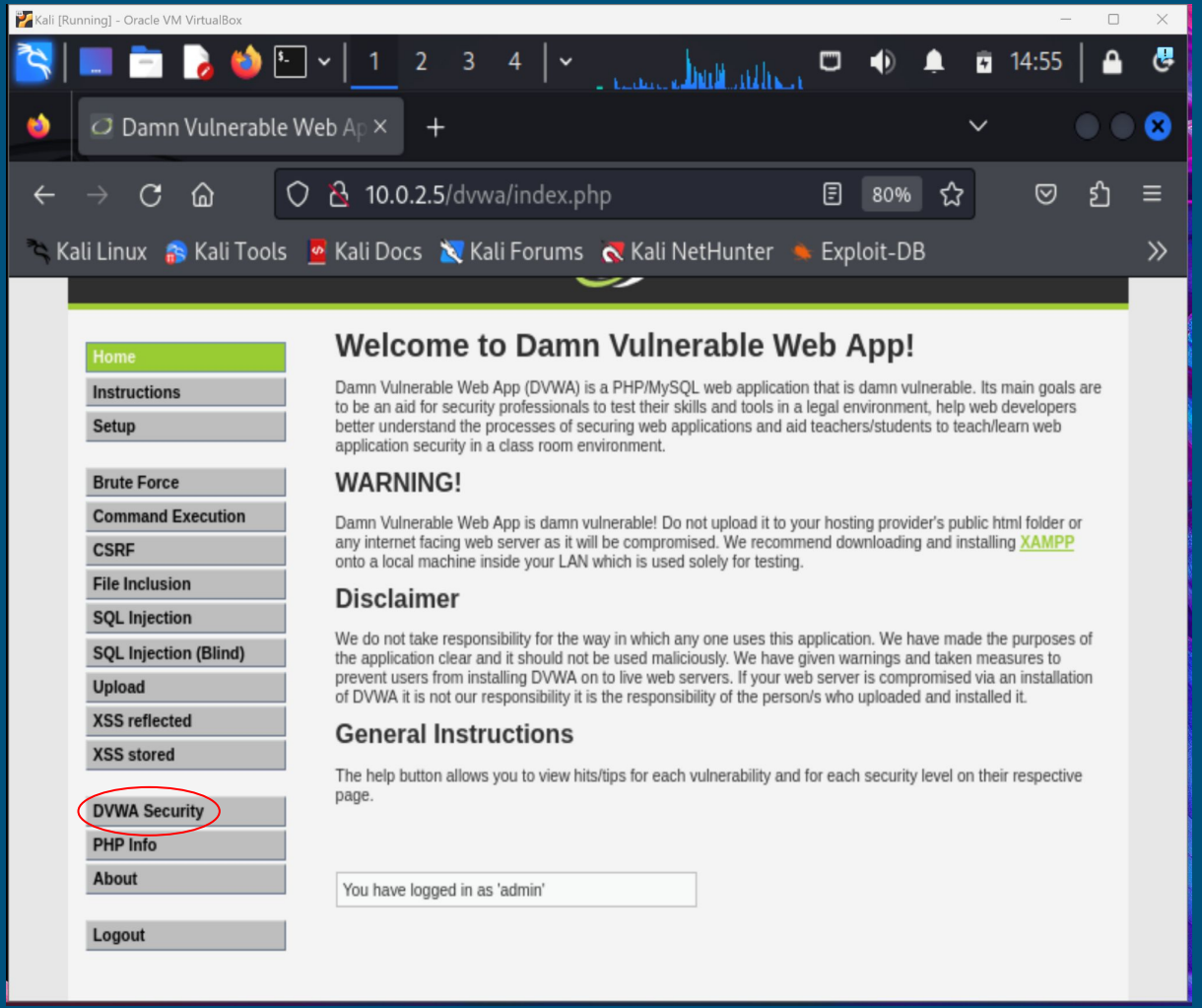

Metasploitable [Running] - Oracle VM VirtualBox
* Starting deferred execution scheduler atd [OK]
* Starting periodic command scheduler crond [OK]
* Starting Tomcat servlet engine tomcat5.5 [OK]
* Starting web server apache2 [OK]
* Running local boot scripts (/etc/rc.local)
nohup: appending output to `nohup.out'
nohup: appending output to `nohup.out'
[OK]
Warning: Never expose this VM to an untrusted network!
Contact: msfdev[at]metasploit.com
Login with msfadmin/msfadmin to get started
metasploitable login: _
Kali [Running] - Oracle VM VirtualBox
kali
us
03 Sep, 14:50
Enter your password
Cancel
Log In
KALI
[Optical Drive] Empty


Metasploitable [Running] - Oracle VM VirtualBox
Last login: Mon Sep 2 19:11:53 EDT 2024 on tty1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
inet6 ::1/128 scope host
valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
link/ether 08:00:27:6c:f5:63 brd ff:ff:ff:ff:ff:ff
inet 10.0.2.5/24 brd 10.0.2.255 scope global eth0
inet6 fe80::a00:27ff:fe6c:f563/64 scope link
valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$
Kali [Running] - Oracle VM VirtualBox
14:53
Metasploitable2 - Linux
Metasploitable2 - Linux
10.0.2.5
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
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
[Optical Drive] Empty

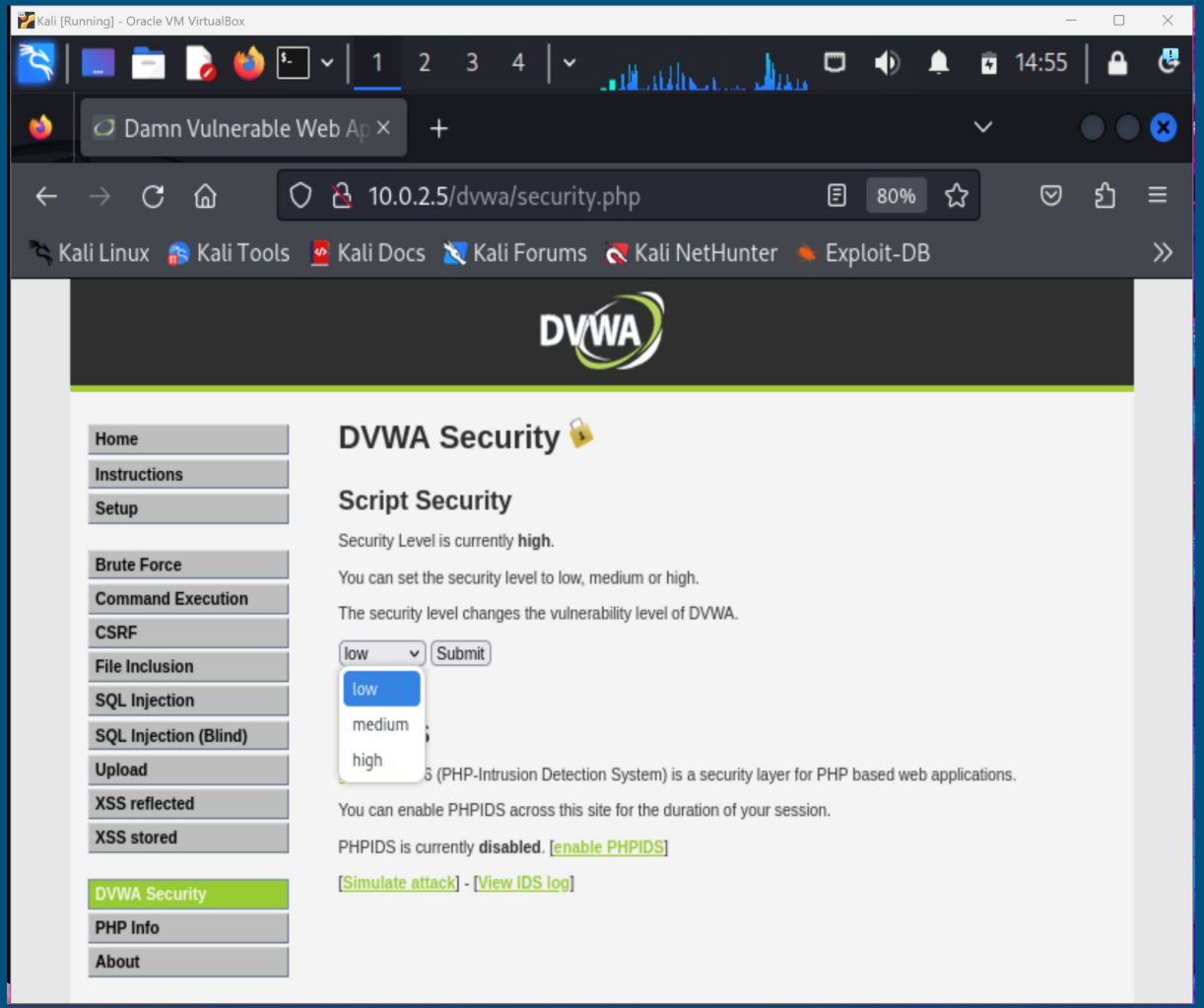
- Log in to DVWA
- Default username: **admin**
- Default password: **password**



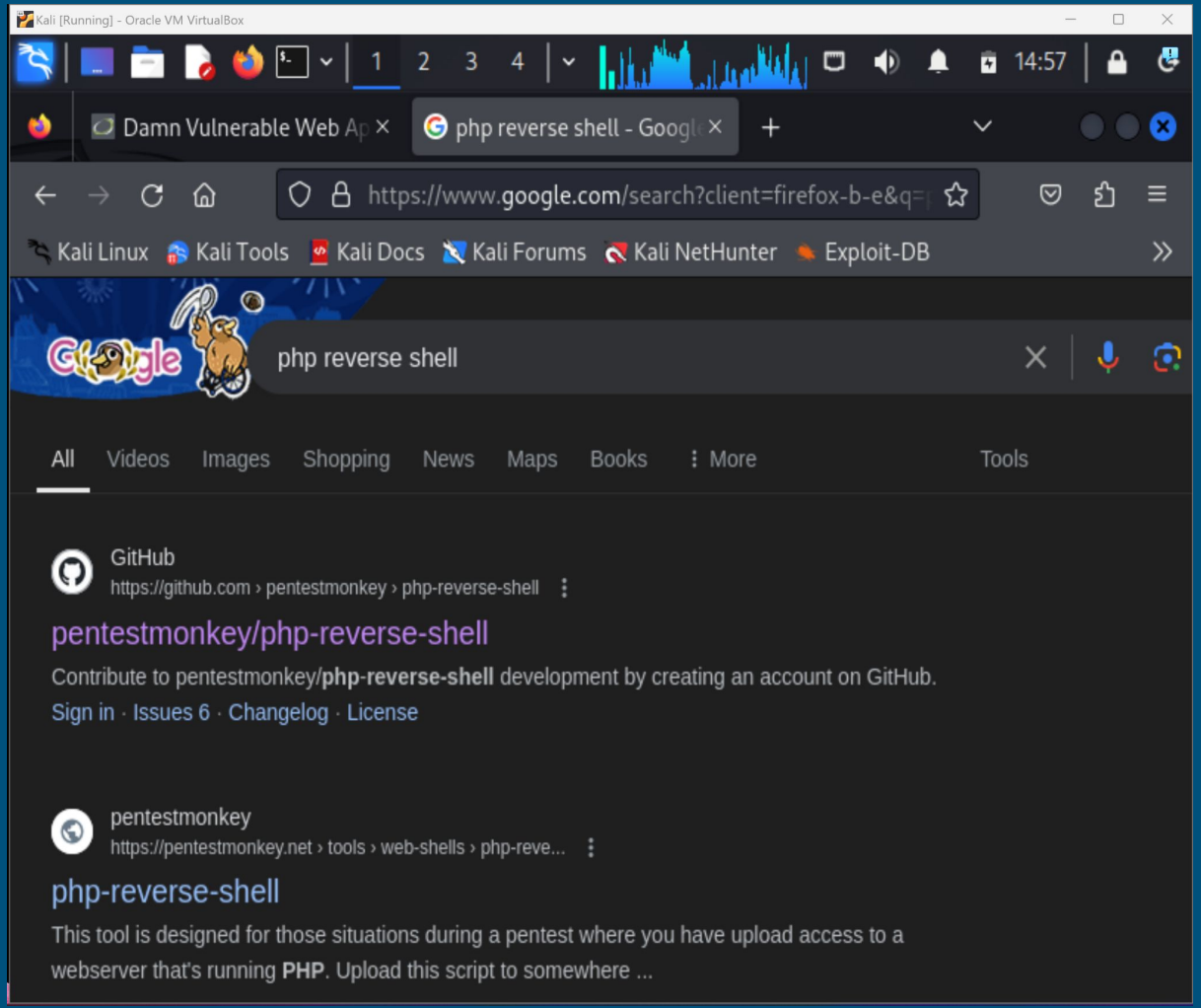
- Acknowledge the variety of services provided by DVWA
- Navigate to the DVWA Security tab



- Change difficulty to low and hit submit
- This will change the security settings to be extremely flimsy
- This is to highlight the ease an attack like this can be done without preventative measures



- Next we are going to acquire a payload to execute on the target
- This time we will use a PHP reverse shell
- The github for pentestmonkey has a good shell to use so we will download it onto the Kali machine



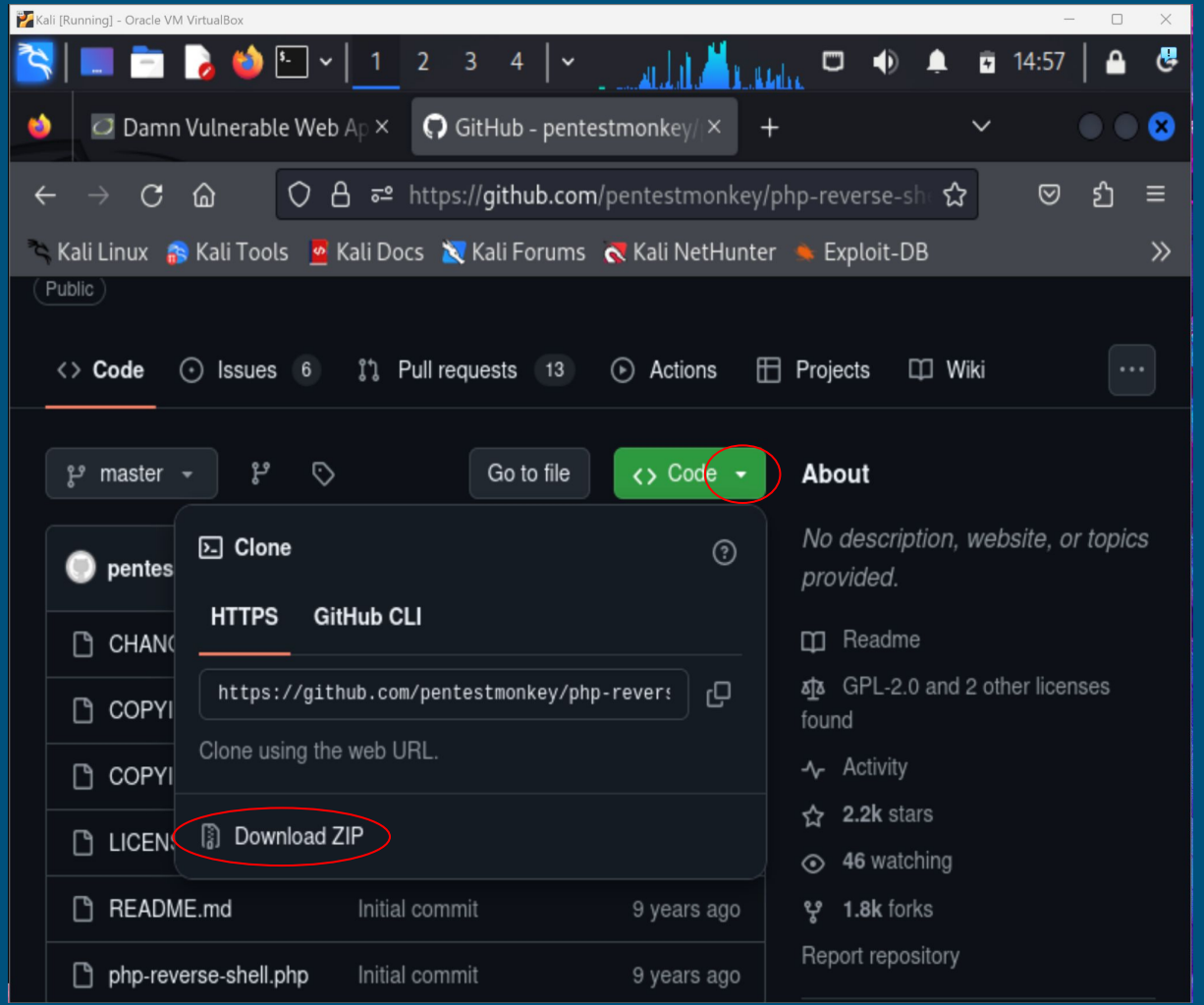
Index

- A payload in this context can be used to refer to any malicious code that can cause harm to the target

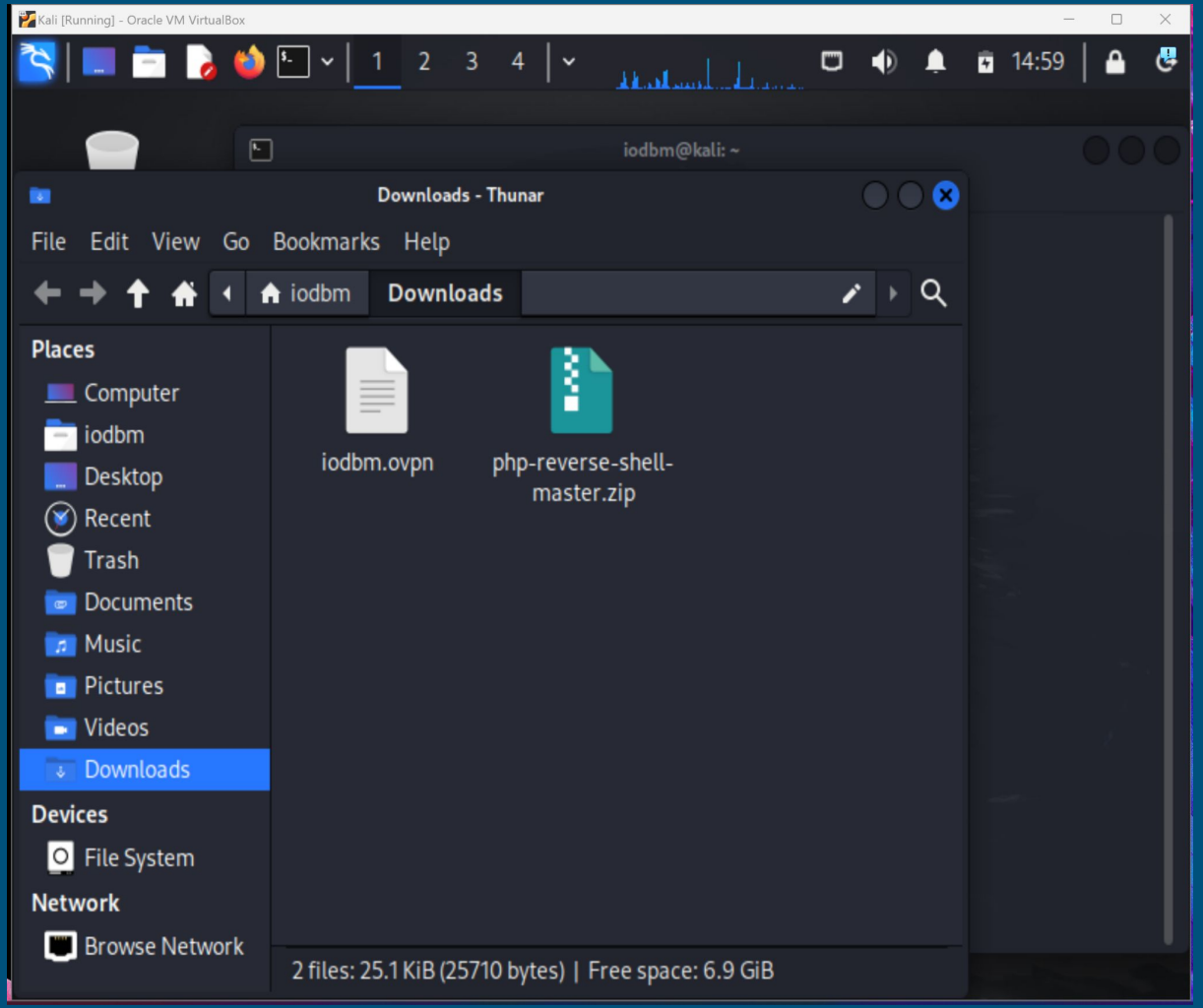
- PHP is a scripting language commonly used with web development which makes it suitable to attack web applications

- A reverse shell is a type of attack where a threat actor connects a target machine to their machine, generally granting them access

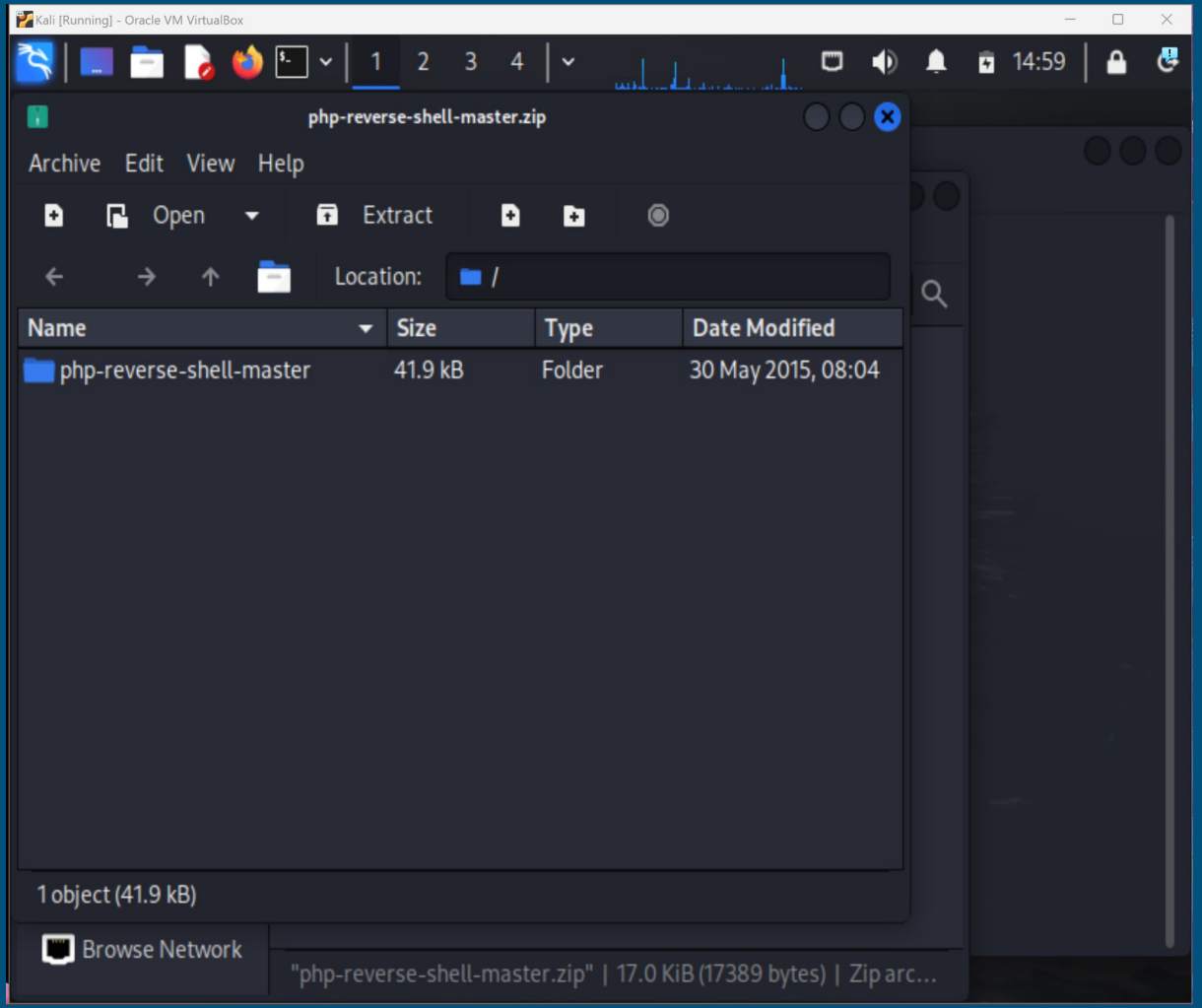
- Once in the site we can click the arrow on the green code button
- Then from that menu we click the Download ZIP button
- This will begin the download



- Navigate to your downloads folder and locate the zip
- Open it

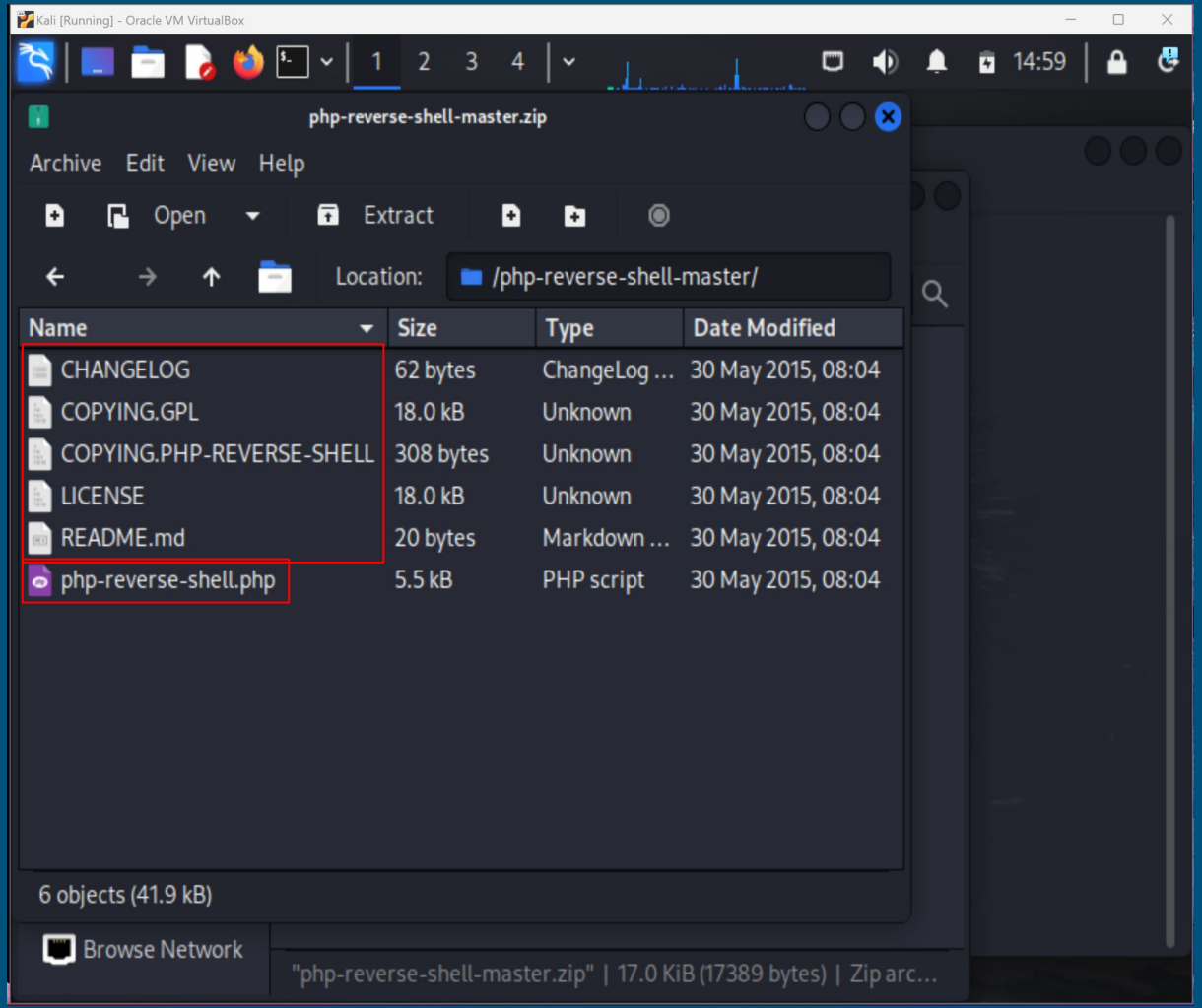


- Inside will be a folder
- We will open that as well

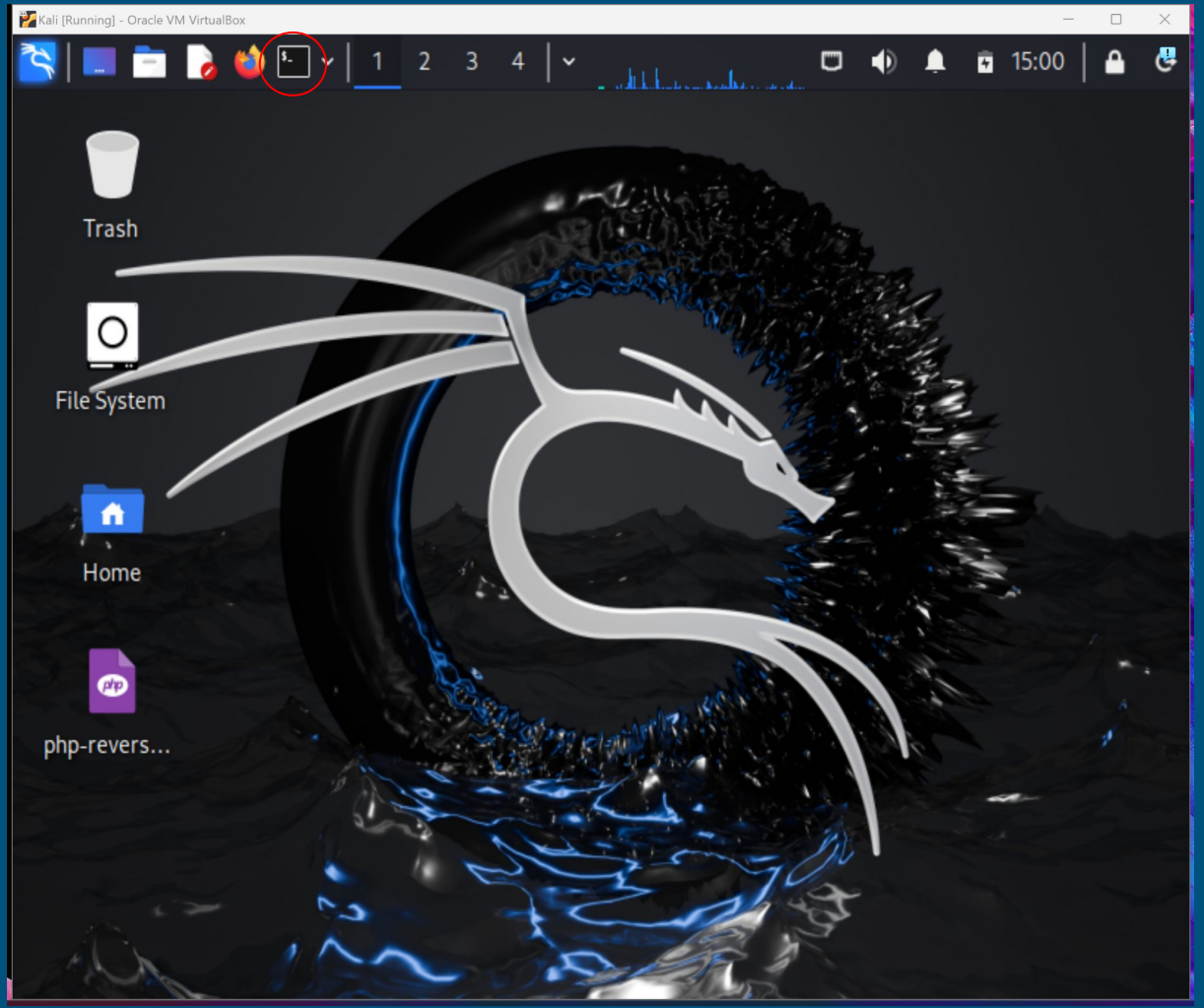


- There will be several documents which correspond with the program

- We will ignore these and instead focus on the purple .php file, this is the payload

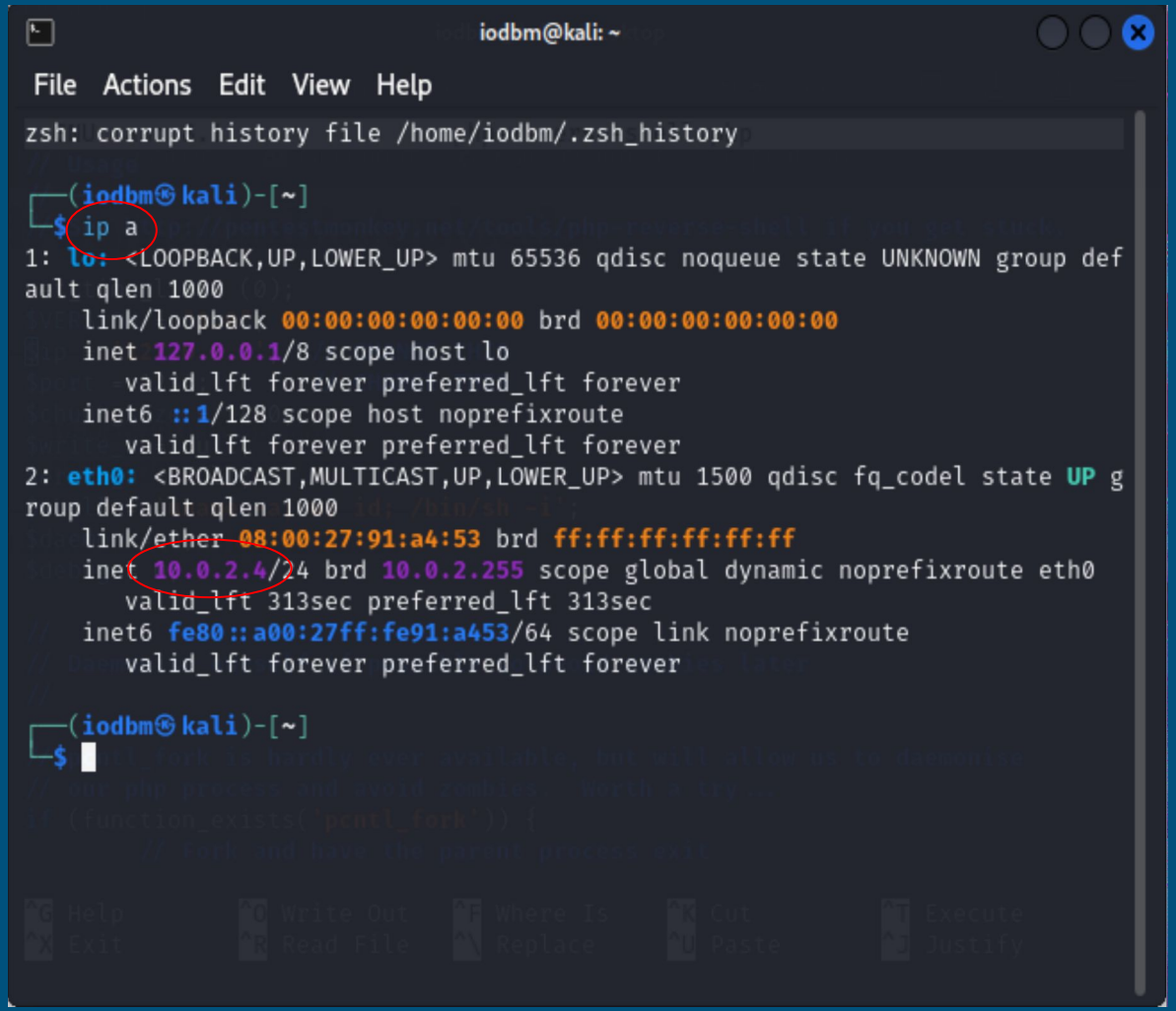


- I chose to move this to my desktop for ease of use
- Next we need to open a new terminal window

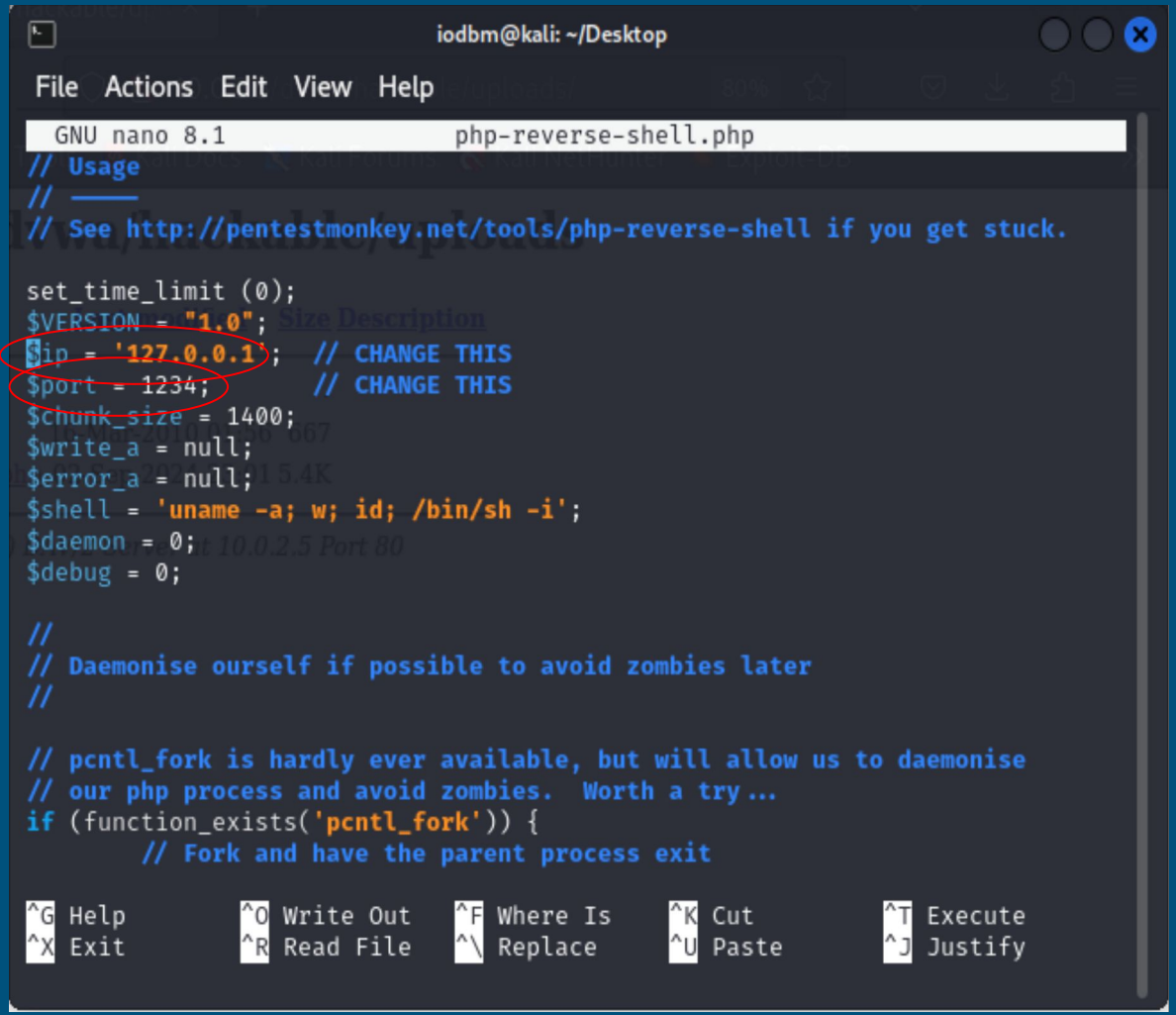


- We can use the “ip a” command again here to find the attacking (Kali) machines IP

- In this case it is 10.0.2.4

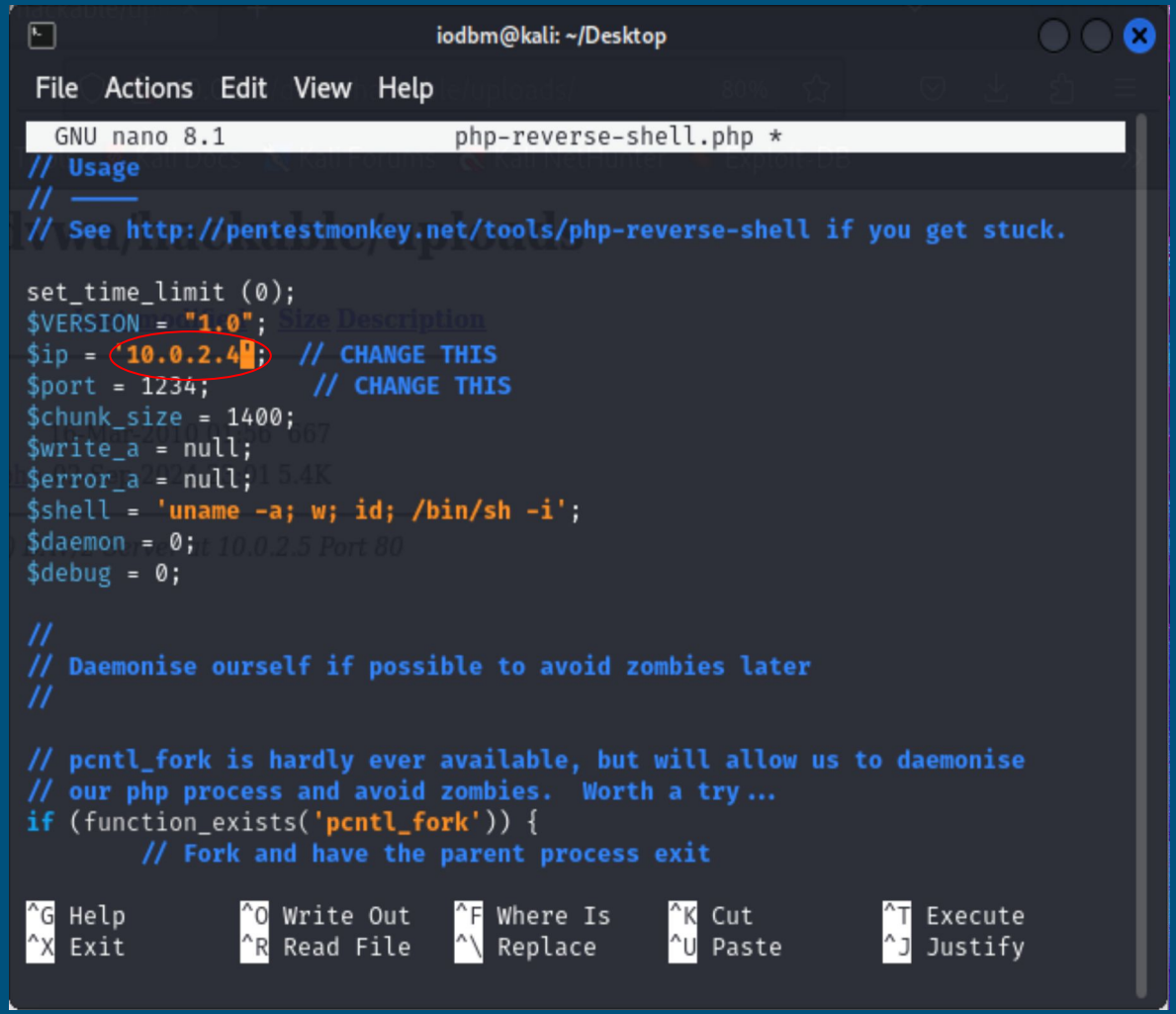


- Now that you have the payload you will need to modify it to make it work on your system
- Change the value of “$ip =” to your attacking (Kali) machines IP
- The port can remain as 1234

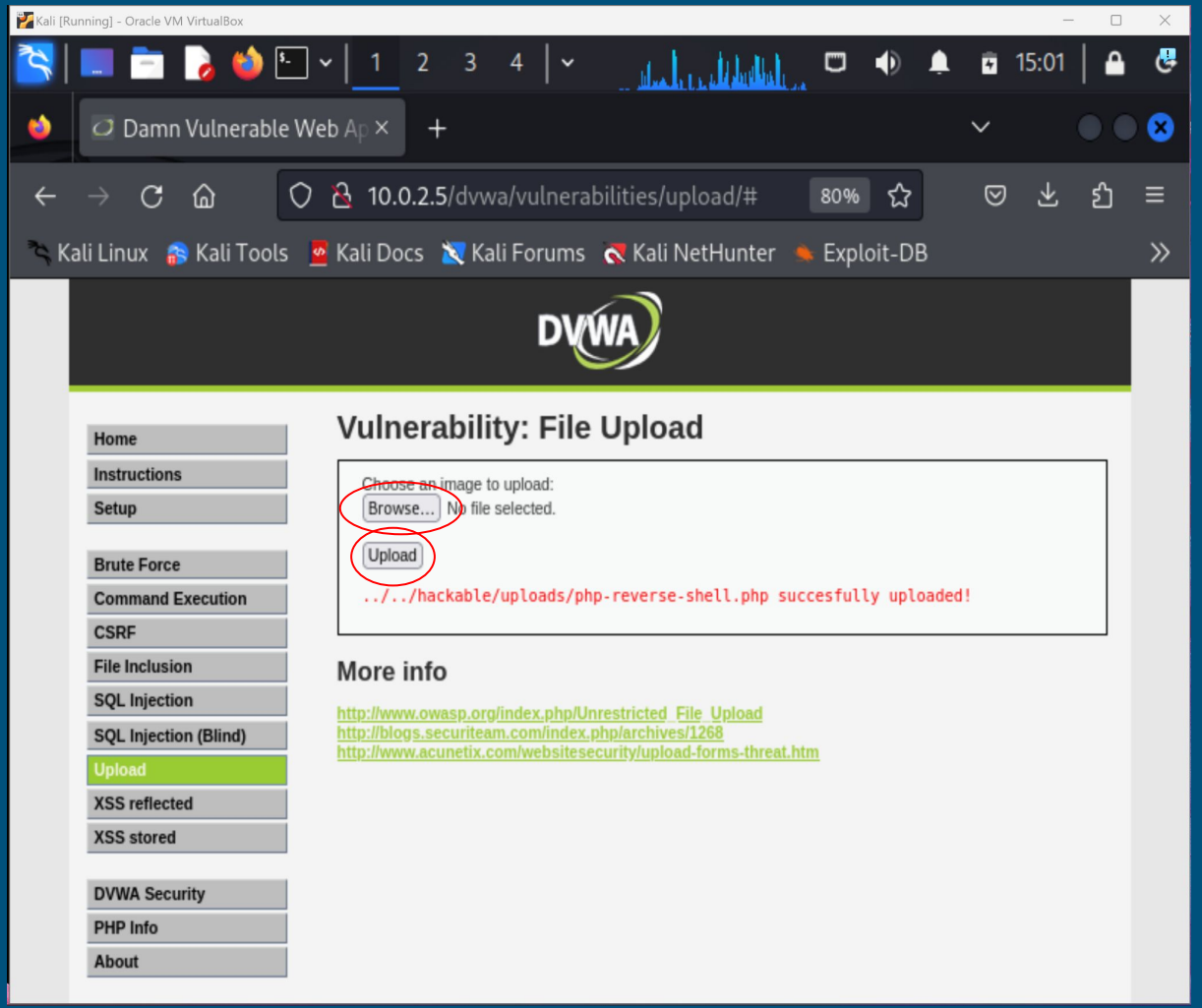


- We have updated the ip

- Now we can Ctrl+x to save our changes and exit



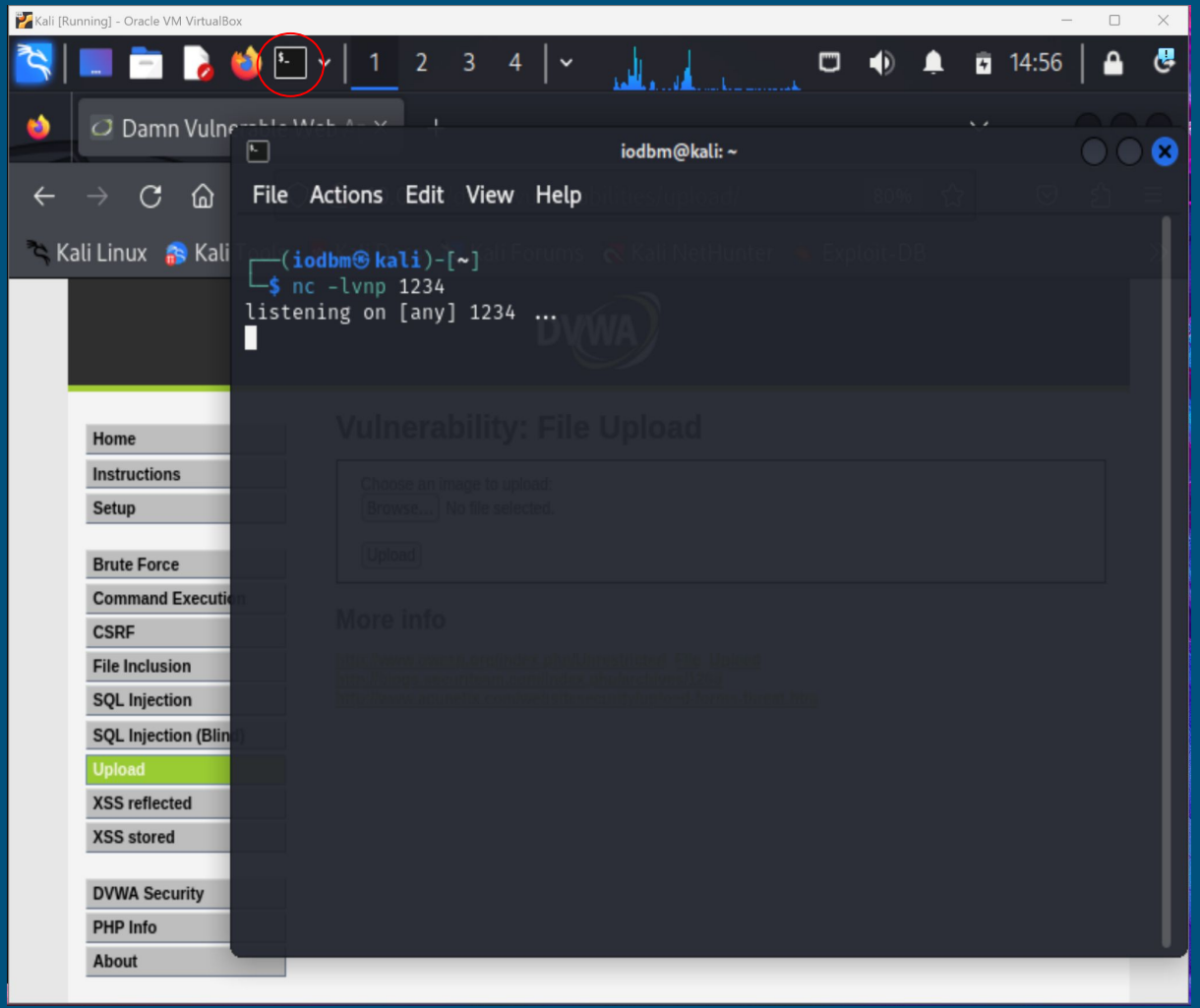
- Now we can move back to the browser and click browse
- Select the file from your desktop (or wherever you stored it)
- Hit open to select it in the browser
- Finally hit upload



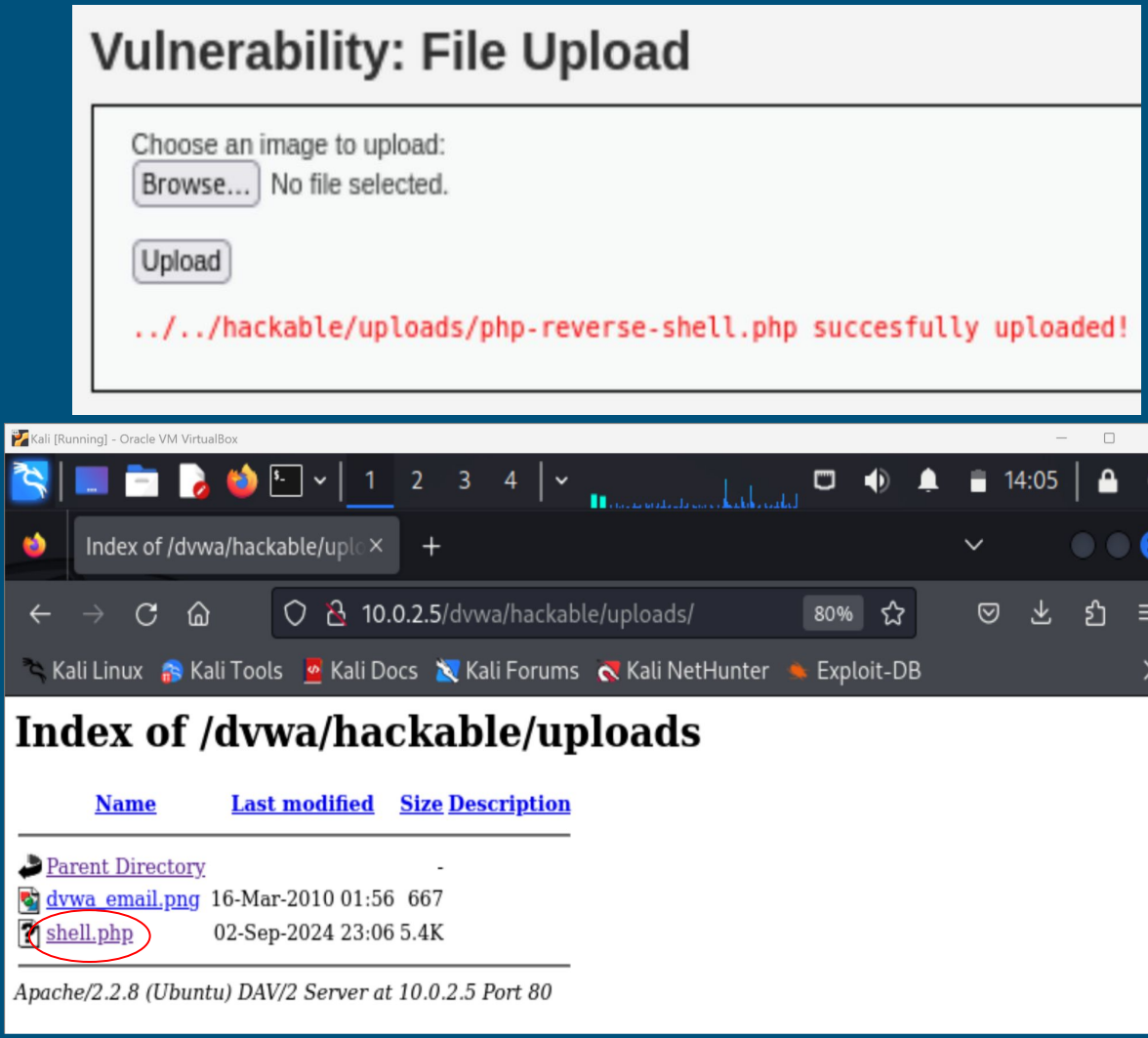
- We then open a new terminal window
- Type the following command:

```
nc -lvnp 1234
```

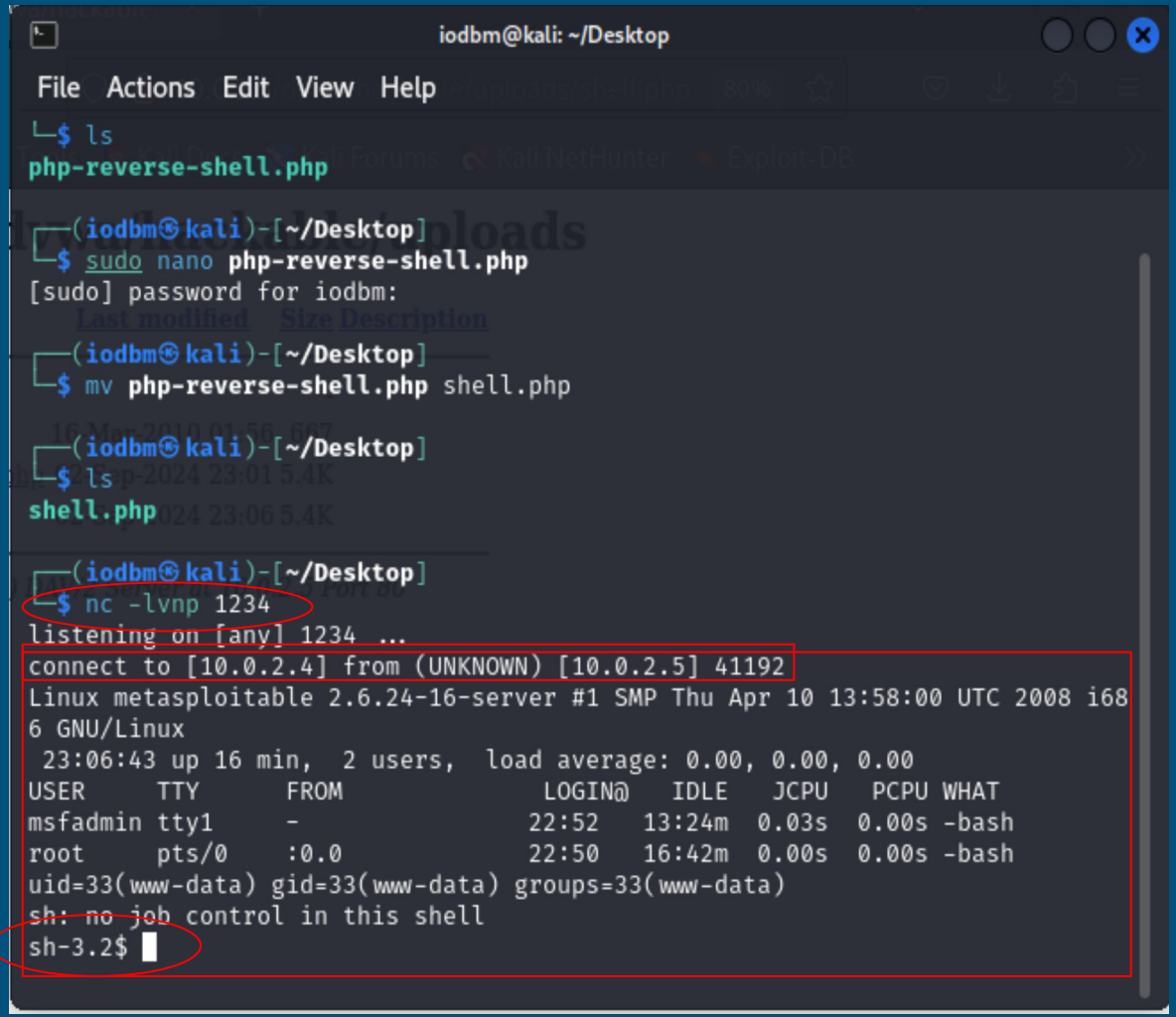
- This will use Netcat (nc) to set up a listener on port 1234



- Now we need to navigate to the directory the file was uploaded to
- You can copy the red text excluding the file name and append it to the end of the current URL
- It will take you to this page
- Now we will click on the php file with our listener active…



- We can see our previous commands for editing the file
- Under that is our listener set up command
- However now we can see several additional lines which show a shell instance has been started



Mitigation

From this scenario

- There are several key points throughout the attack where we can implement defences
- It is important to be aware of all the potential attack vectors of your system so you can properly defend them
- This is what a pentest is designed to find

- Limit the permitted file types and file extensions users can upload

- Securely store all uploaded files to prevent unauthorised access

- Scan all incoming files to detect for malicious code or viruses

How can we prevent this?

- Policies - Improved guidelines for companies and developers
- Tools & Techniques - Secure coding practices
- Training - For employees to recognise threats
- Frameworks - Such as OWASP or NIST

To Conclude

- For every measure in place to prevent or inhibit threat actors, they have a countermeasure or method to bypass it.

- There is no such thing as a perfectly secure system.

- However by putting ourselves in the attackers shoes, we can find vulnerabilities and more effectively prevent exploits

Thank You

Questions?