H4cker

H4CKER.ORG

Ethical Hacking Bootcamp

DAY 1

Safari Live Training by Omar Santos

https://bootcamp.h4cker.org

Introduction

This guide is a collection of exercises for the training “Ethical Hacking Bootcamp with Hands-on Labs” authored and delivered by Omar Santos and delivered through O’Reilly Safari Books Online. For more information about the training visit https://bootcamp.h4cker.org

Training Summary

This course is useful you are starting your cybersecurity career, seeking your Certified Ethical Hacker, Offensive Security Certified Professional (OSCP) Certification, or CompTIA PenTest+ certifications; or just interested in learning more about cyber security, this two-day training session is a great place to start. is a comprehensive Ethical Hacking (security penetration testing) hands-on boot camp! In this 3-day course you will get the training and experience you need to successfully launch your ethical hacking career. You will participate in live discussions, demos, whiteboard instruction and lab exercises. No prior experience is needed. This course provides step-by-step real-life scenarios. You will see first-hand how an ethical hacker performs initial reconnaissance of a victim and how to assess systems and network security controls security posture. This training includes live discussions, demos, whiteboard instruction and screencasts.

Helpful Resources Prior to Taking the Live Training:

- Security Penetration Testing The Art of Hacking Series LiveLessons (video)
- Wireless Networks, IoT, and Mobile Devices Hacking (video)
- Enterprise Penetration Testing and Continuous Monitoring (video)
- Hacking Web Applications (video)
- Security Fundamentals (video)

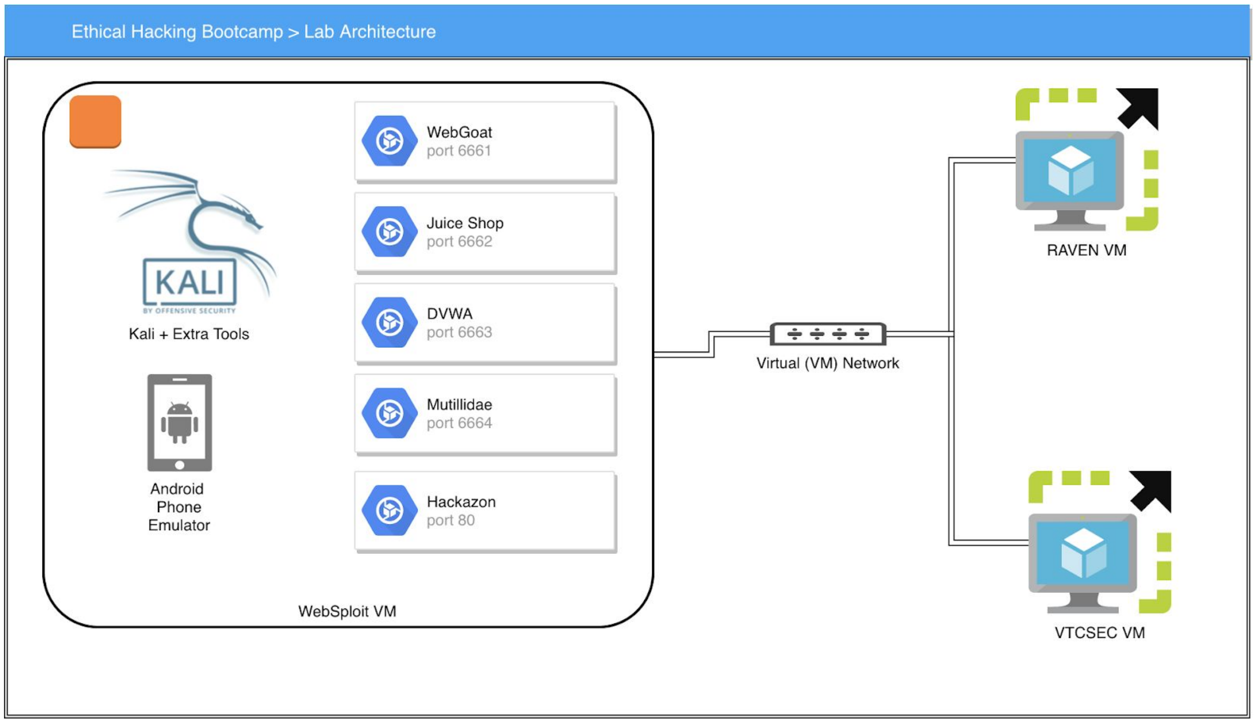
Lab Setup

You can build your own lab as elaborate as you would like. However, for the purpose of this class, the following virtual machines (VMs) will be used.

- WebSploit: Kali + Additional Tools + Vulnerable Applications in Docker containers...
- Raven: A vulnerable VM that you will use to perform a full assessment (from reconnaissance to full compromise).
- VTCSEC: A second vulnerable VM that you will use to perform a full assessment (from reconnaissance to full compromise)

Lab Architecture and Topology

The following is the lab architecture for this class.



Deploying Your Virtual Machines

You can deploy and configure your VMs using Virtual Box, VMWare Workstation Player, VMWare Workstation Pro (Windows), VMWare Fusion (Mac), or vSphere Hypervisor (free ESXi server).

You should create a VM-only network (as shown in the previous figure) to deploy your vulnerable VMs and perform several of the attacks using WebSploit (Kali Linux).

You can configure a separate network interface in your WebSploit VM to connect to the rest of your network and subsequently the Internet.

Penetration Testing Linux Distributions

Several Linux distributions package numerous penetration testing tools. The purpose of these Linux distributions is to make it easier for individuals to get started with penetration testing, without having to worry about software dependencies and compatibility issues that could be introduced when installing and deploying such tools. The following are the most popular penetration testing Linux distributions:

Kali Linux

You can download Kali Linux from https://www.kali.org. Offensive Security released a free open source book and course about how to install, customize, and use Kali Linux. The book and the course can be accessed at https://kali.training.

Parrot Security OS

You can download Parrot from https://www.parrotsec.org and access the documentation at https://docs.parrotsec.org.

Black Arch

You can download BlackArch Linux from https://blackarch.org and access the documentation at https://blackarch.org/guide.html. BlackArch Linux source code can be accessed at https://github.com/BlackArch/blackarch.

Pentoo

You can download Pentoo at: https://www.pentoo.ch

The following resource provides additional details about these distributions and several additional ones that you can use to build your own labs:

https://h4cker.org/go/distros

Exercise 1.0: Kali Linux Top Post Install Customizations and Tips

As per the instructions from Offensive Security here, "VMware recommends using the distribution-specific open-vm-tools (OVT) instead of the VMware Tools package for guest machines. To install open-vm-tools in Kali, first make sure you are fully updated, and then enter the following:"

```
apt update && apt -y full-upgrade

# Reboot now in case you have updated to a new kernel. Once rebooted:

apt -y --reinstall install open-vm-tools-desktop fuse
reboot
```

(Optional) You can refer to the instructions here, for adding support for Shared Folders when using OVT.

Refer to the following link for additional post install tips including:
https://www.offensive-security.com/kali-linux/top-10-post-install-tips

- enabling or disabling the Intelligent Sidebar option
- adding your SSH public key
- installing the NVIDIA drivers if you need them
- disabling the Screen Lock Gnome feature
- adding a non-root User if you're not comfortable running as root
- keeping the Kali system up to date

Passive Reconnaissance

Exercise 2.1: Passive Recon Finding Information About You

1. Using your own system (i.e., desktop, laptop, or Kali Linux VM, etc.) find your public IP address and lookup all information you can about it.
2. Try whatsmyip.org or ipchicken.com

3. Pick a random company or institution that is not a Fortune 500 or a big Internet service provider (ISP), so that makes it harder for you to find information about it. Do NOT launch any active recon against such organization.
4. You can use Maltego for this. Watch this video if you are not familiar with Maltego.
5. Also use recon-ng and the harvester. If you want a refresher on how to use these tools, watch this video and this one.
   a. Find any information you can about network infrastructure
   b. Registered domain names
   c. IP Address allocations
   d. Open services and banners
   e. Leverage the tools listed at: https://h4cker.org/recon

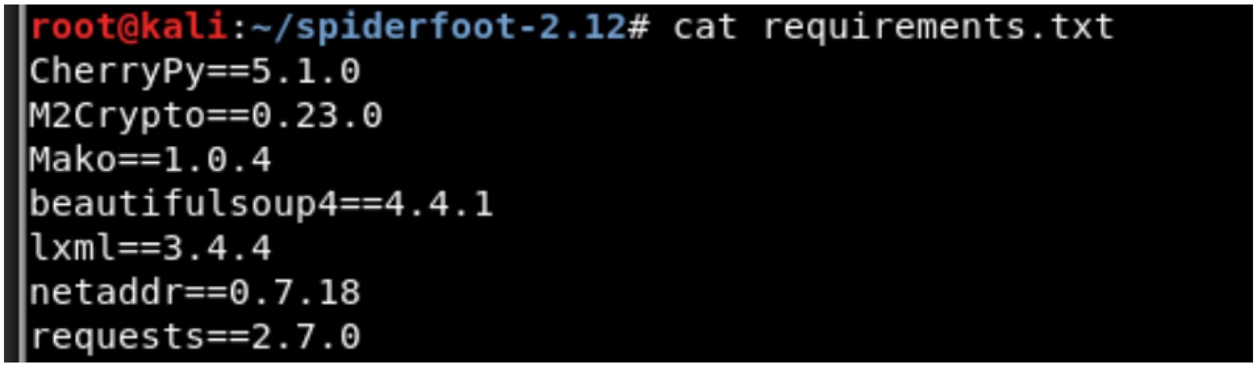
Exercise 2.2: SpiderFoot

- Download SpiderFoot from: https://www.spiderfoot.net/download/
- To install SpiderFoot you only need to un-targz the package, as follows:

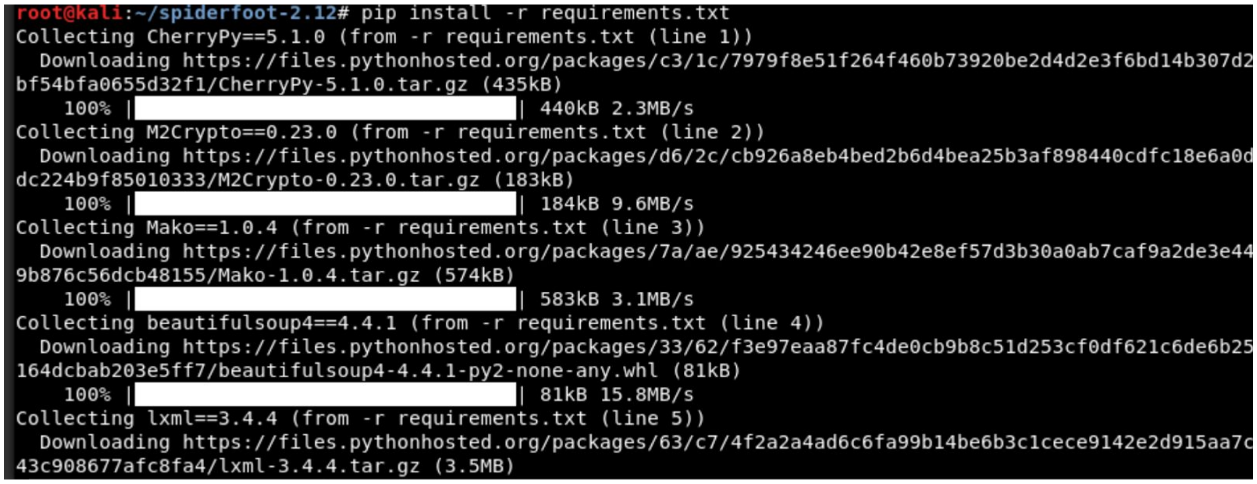
```
# tar zxvf spiderfoot-X.X.X-src.tar.gz
# cd spiderfoot-X.X.X
```

- Then make sure that all the necessary python modules are installed by using the command below and demonstrated in the following figures:

```
# pip install  -r requirements.txt
```

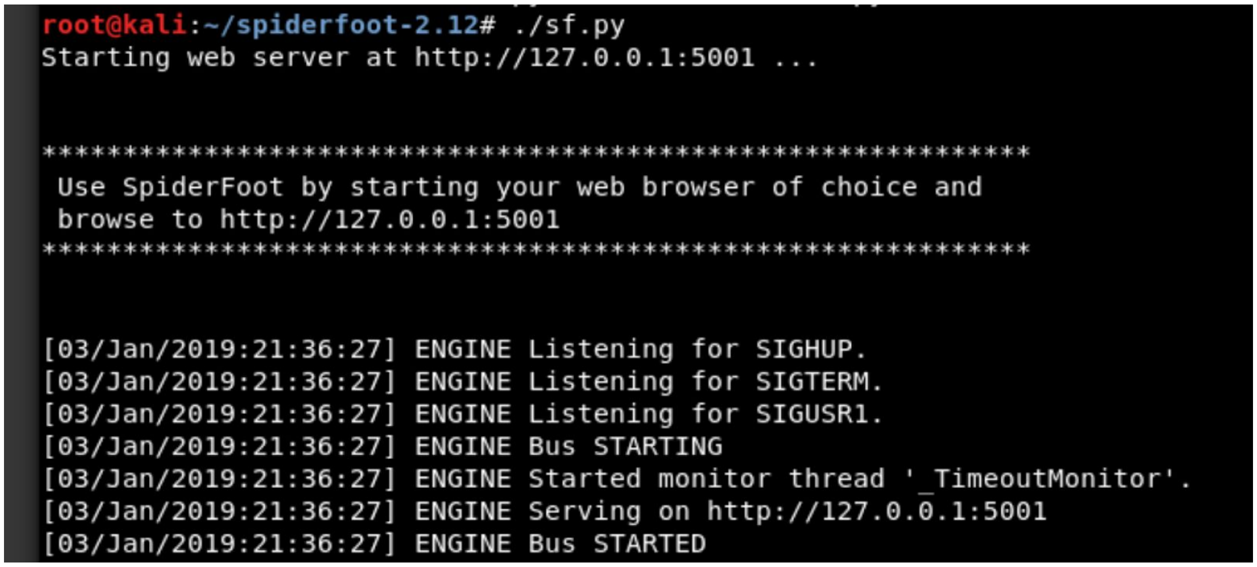


```
root@kali:~/spiderfoot-2.12# cat requirements.txt
CherryPy==5.1.0
M2Crypto==0.23.0
Mako==1.0.4
beautifulsoup4==4.4.1
lxml==3.4.4
netaddr==0.7.18
requests==2.7.0
```



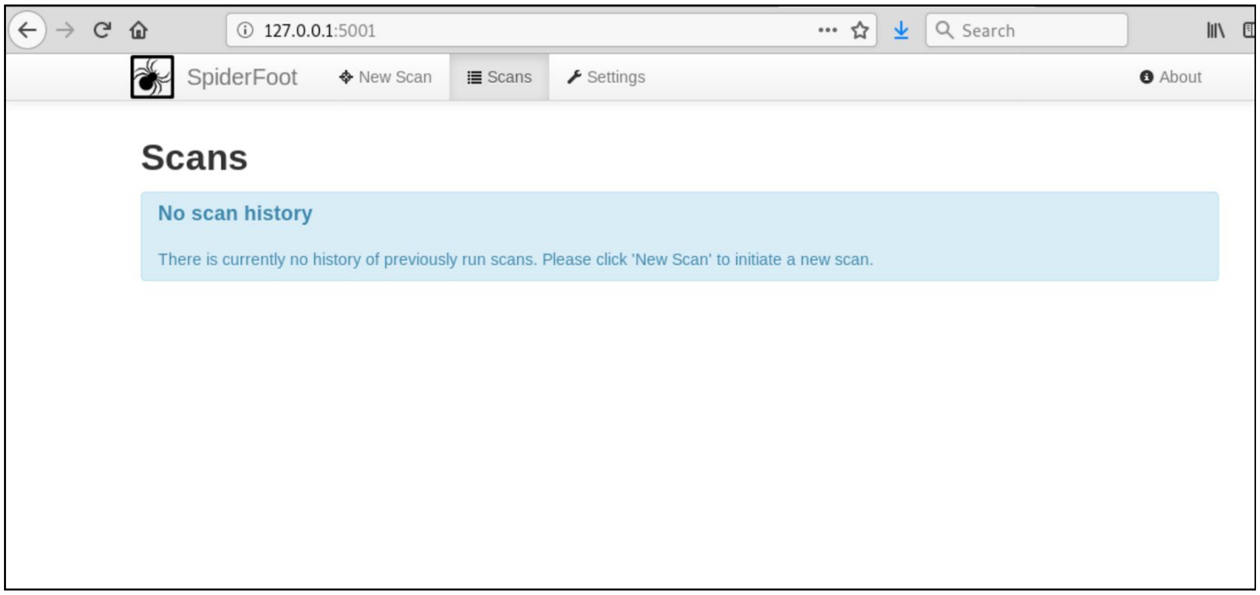
```
root@kali:~/spiderfoot-2.12# pip install -r requirements.txt
Collecting CherryPy==5.1.0 (from -r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/c3/1c/7979f8e51f264f460b73920be2d4d2e3f6bd14b307d2
bf54bfa0655d32f1/CherryPy-5.1.0.tar.gz (435kB)
    100% |████████████████████████████████| 440kB 2.3MB/s
Collecting M2Crypto==0.23.0 (from -r requirements.txt (line 2))
  Downloading https://files.pythonhosted.org/packages/d6/2c/cb926a8eb4bed2b6d4bea25b3af898440cdfc18e6a0d
dc224b9f85010333/M2Crypto-0.23.0.tar.gz (183kB)
    100% |████████████████████████████████| 184kB 9.6MB/s
Collecting Mako==1.0.4 (from -r requirements.txt (line 3))
  Downloading https://files.pythonhosted.org/packages/7a/ae/925434246ee90b42e8ef57d3b30a0ab7caf9a2de3e44
9b876c56dcb48155/Mako-1.0.4.tar.gz (574kB)
    100% |████████████████████████████████| 583kB 3.1MB/s
Collecting beautifulsoup4==4.4.1 (from -r requirements.txt (line 4))
  Downloading https://files.pythonhosted.org/packages/33/62/f3e97eaa87fc4de0cb9b8c51d253cf0df621c6de6b25
164dcbab203e5ff7/beautifulsoup4-4.4.1-py2-none-any.whl (81kB)
    100% |████████████████████████████████| 81kB 15.8MB/s
Collecting lxml==3.4.4 (from -r requirements.txt (line 5))
  Downloading https://files.pythonhosted.org/packages/63/c7/4f2a2a4ad6c6fa99b14be6b3c1cece9142e2d915aa7c
43c908677afc8fa4/lxml-3.4.4.tar.gz (3.5MB)
```

- Start SpiderFoot by using the `sf.py` command as shown below:



```
root@kali:~/spiderfoot-2.12# ./sf.py
Starting web server at http://127.0.0.1:5001 ...


*************************************************************
 Use SpiderFoot by starting your web browser of choice and
 browse to http://127.0.0.1:5001
*************************************************************


[03/Jan/2019:21:36:27] ENGINE Listening for SIGHUP.
[03/Jan/2019:21:36:27] ENGINE Listening for SIGTERM.
[03/Jan/2019:21:36:27] ENGINE Listening for SIGUSR1.
[03/Jan/2019:21:36:27] ENGINE Bus STARTING
[03/Jan/2019:21:36:27] ENGINE Started monitor thread '_TimeoutMonitor'.
[03/Jan/2019:21:36:27] ENGINE Serving on http://127.0.0.1:5001
[03/Jan/2019:21:36:27] ENGINE Bus STARTED
```

- Open the browser and go to http://127.0.0.1:5001 to access the SpiderFoot GUI.
- When you run SpiderFoot for the first time, there is no historical data, so you should be presented with a screen like the following:

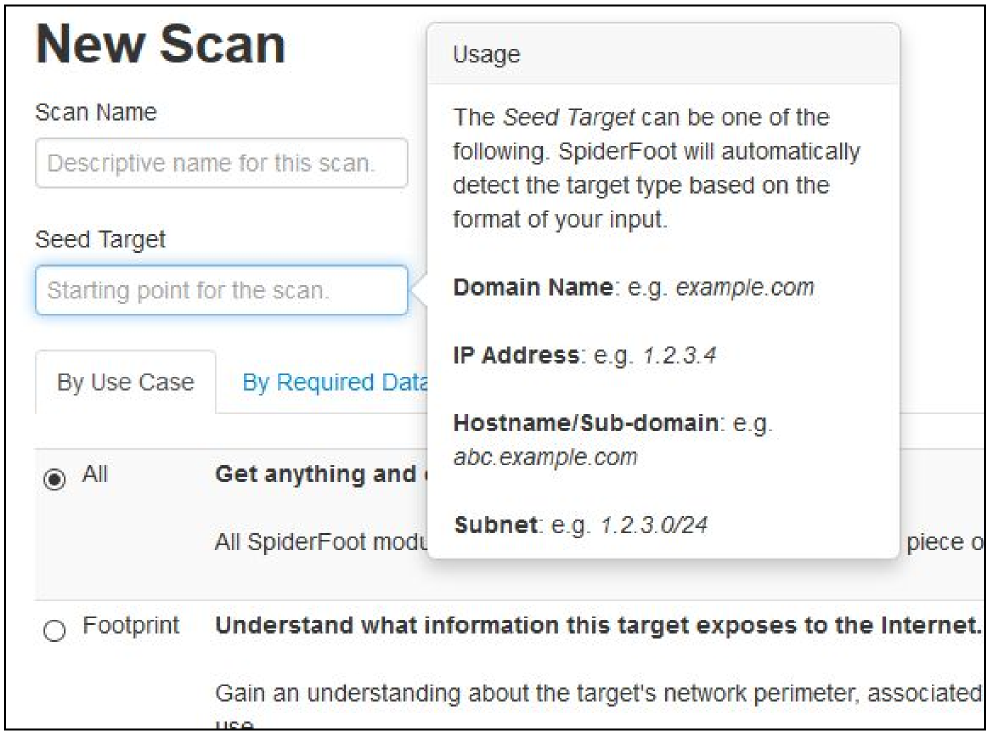


Scans

No scan history

There is currently no history of previously run scans. Please click 'New Scan' to initiate a new scan.

- Click on the ‘New Scan’ button in the top menu bar. You will then need to define a name for your scan (these are non-unique) and a target (also non-unique):



New Scan

Scan Name

Descriptive name for this scan.

Seed Target

Starting point for the scan.

Usage

The *Seed Target* can be one of the following. SpiderFoot will automatically detect the target type based on the format of your input.

Domain Name: e.g. *example.com*

IP Address: e.g. *1.2.3.4*

Hostname/Sub-domain: e.g. *abc.example.com*

Subnet: e.g. *1.2.3.0/24*

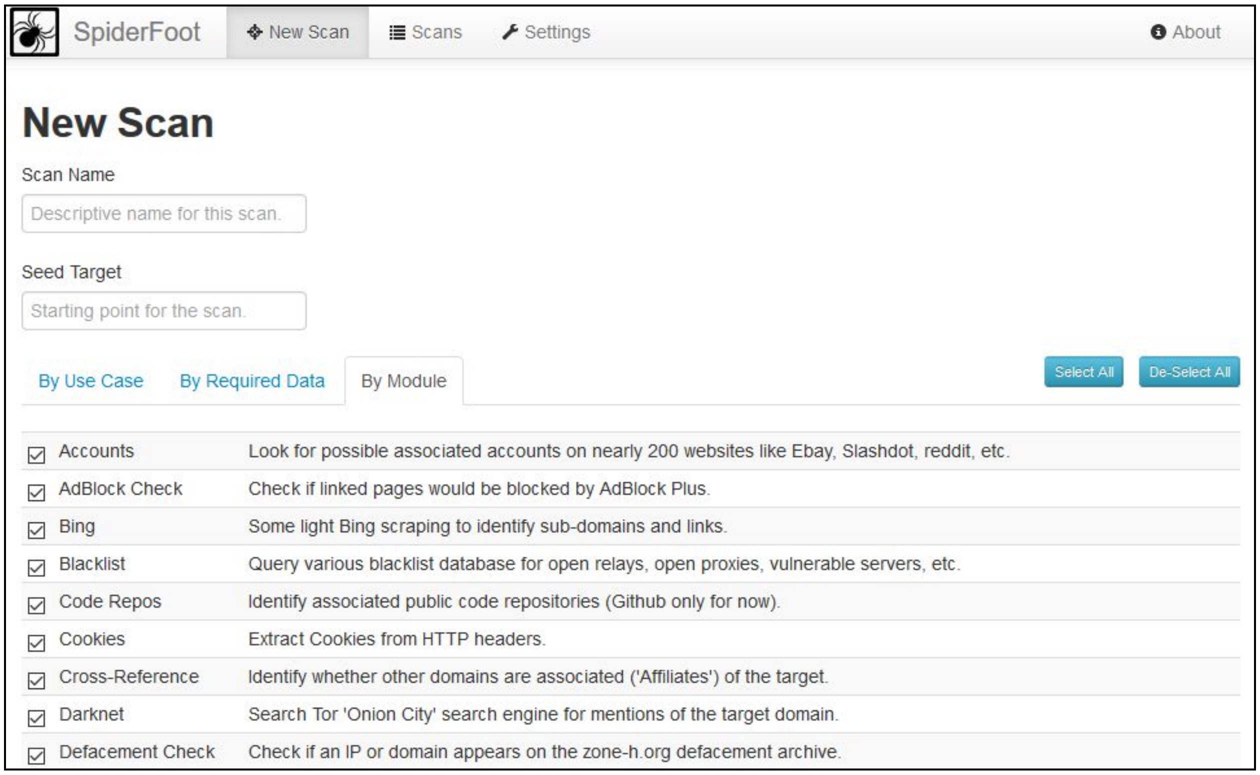
(●) All **Get anything and**

All SpiderFoot modu ... piece o

( ) Footprint **Understand what information this target exposes to the Internet.**

Gain an understanding about the target's network perimeter, associated

- You can then define how you would like to run the scan - either by use case (the tab selected by default), by data required or by module.
- Module-based scanning is for more advanced users who are familiar with the behavior and data provided by different modules, and want more control over the scan:



New Scan

Scan Name

Descriptive name for this scan.

Seed Target

Starting point for the scan.

☑	Accounts	Look for possible associated accounts on nearly 200 websites like Ebay, Slashdot, reddit, etc.
☑	AdBlock Check	Check if linked pages would be blocked by AdBlock Plus.
☑	Bing	Some light Bing scraping to identify sub-domains and links.
☑	Blacklist	Query various blacklist database for open relays, open proxies, vulnerable servers, etc.
☑	Code Repos	Identify associated public code repositories (Github only for now).
☑	Cookies	Extract Cookies from HTTP headers.
☑	Cross-Reference	Identify whether other domains are associated ('Affiliates') of the target.
☑	Darknet	Search Tor 'Onion City' search engine for mentions of the target domain.
☑	Defacement Check	Check if an IP or domain appears on the zone-h.org defacement archive.

- Tip: If you select a module that depends on event types only provided by other modules, but those modules are not selected, you will get no results.
- Run your scan. Choose any arbitrary victim (your own website, your own name, etc.)

Exercise 2.3: Sublist3r

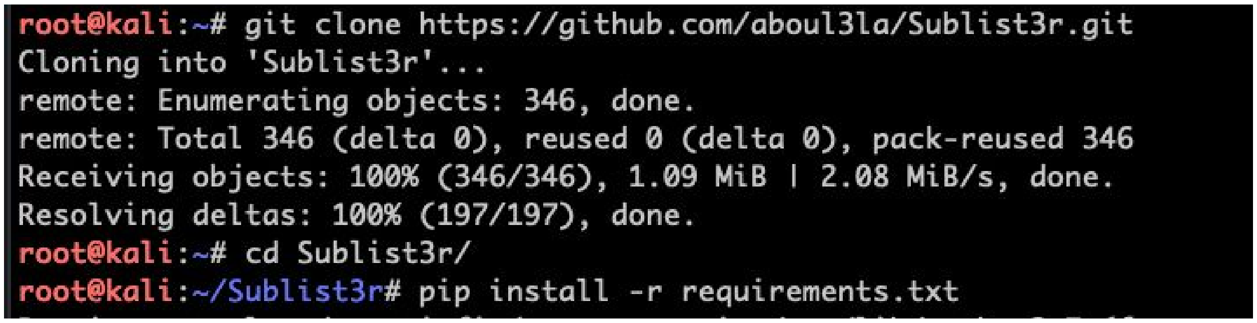
As it reads in their GitHub repository "Sublist3r is a python tool designed to enumerate subdomains of websites using OSINT. It helps penetration testers and bug hunters collect and gather subdomains for the domain they are targeting. Sublist3r enumerates subdomains using many search engines such as Google, Yahoo, Bing, Baidu, and Ask. Sublist3r also enumerates subdomains using Netcraft, Virustotal, ThreatCrowd, DNSdumpster, and ReverseDNS.
subbrute was integrated with Sublist3r to increase the possibility of finding more subdomains using bruteforce with an improved wordlist. The credit goes to TheRook who is the author of subbrute."

1. Download Sublist3r:

```
# git clone https://github.com/aboul3la/Sublist3r.git
```

2. Install the dependencies/requirements by using `pip`:

```
# sudo pip install -r requirements.txt
```

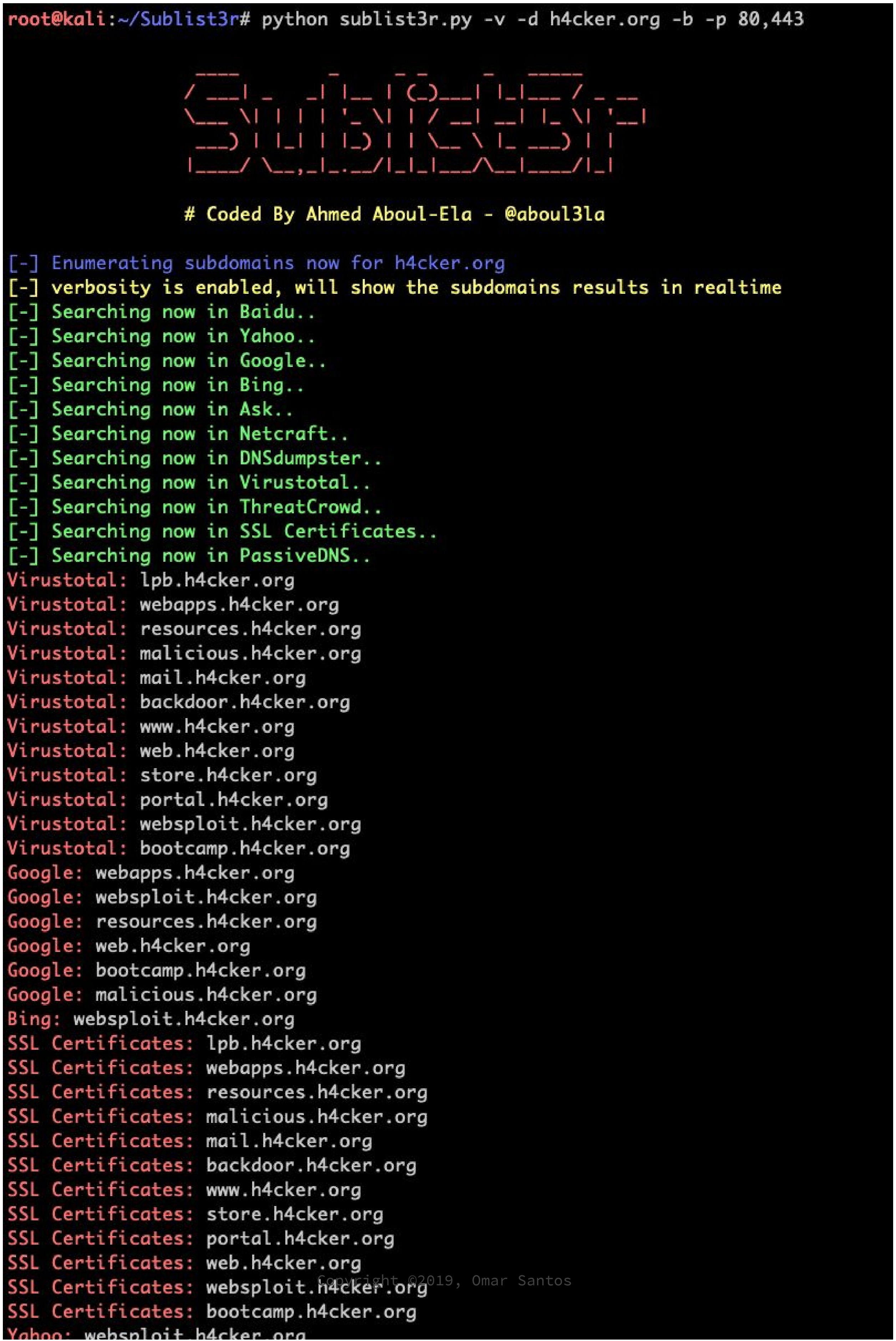


```
root@kali:~# git clone https://github.com/aboul3la/Sublist3r.git
Cloning into 'Sublist3r'...
remote: Enumerating objects: 346, done.
remote: Total 346 (delta 0), reused 0 (delta 0), pack-reused 346
Receiving objects: 100% (346/346), 1.09 MiB | 2.08 MiB/s, done.
Resolving deltas: 100% (197/197), done.
root@kali:~# cd Sublist3r/
root@kali:~/Sublist3r# pip install -r requirements.txt
```

3. Run Sublist3r to your own domain.

```
# python sublist3r.py -v -d your-domain.com -b -p 80,443
```

The following example is against the h4cker.org domain.



```
root@kali:~/Sublist3r# python sublist3r.py -v -d h4cker.org -b -p 80,443

                 ____        _     _ _     _   _____
                / ___| _   _| |__ | (_)___| |_|___ / _ __
                \___ \| | | | '_ \| | / __| __| |_ \| '__|
                 ___) | |_| | |_) | | \__ \ |_ ___) | |
                |____/ \__,_|_.__/|_|_|___/\__|____/|_|

                # Coded By Ahmed Aboul-Ela - @aboul3la

[-] Enumerating subdomains now for h4cker.org
[-] verbosity is enabled, will show the subdomains results in realtime
[-] Searching now in Baidu..
[-] Searching now in Yahoo..
[-] Searching now in Google..
[-] Searching now in Bing..
[-] Searching now in Ask..
[-] Searching now in Netcraft..
[-] Searching now in DNSdumpster..
[-] Searching now in Virustotal..
[-] Searching now in ThreatCrowd..
[-] Searching now in SSL Certificates..
[-] Searching now in PassiveDNS..
Virustotal: lpb.h4cker.org
Virustotal: webapps.h4cker.org
Virustotal: resources.h4cker.org
Virustotal: malicious.h4cker.org
Virustotal: mail.h4cker.org
Virustotal: backdoor.h4cker.org
Virustotal: www.h4cker.org
Virustotal: web.h4cker.org
Virustotal: store.h4cker.org
Virustotal: portal.h4cker.org
Virustotal: websploit.h4cker.org
Virustotal: bootcamp.h4cker.org
Google: webapps.h4cker.org
Google: websploit.h4cker.org
Google: resources.h4cker.org
Google: web.h4cker.org
Google: bootcamp.h4cker.org
Google: malicious.h4cker.org
Bing: websploit.h4cker.org
SSL Certificates: lpb.h4cker.org
SSL Certificates: webapps.h4cker.org
SSL Certificates: resources.h4cker.org
SSL Certificates: malicious.h4cker.org
SSL Certificates: mail.h4cker.org
SSL Certificates: backdoor.h4cker.org
SSL Certificates: www.h4cker.org
SSL Certificates: store.h4cker.org
SSL Certificates: portal.h4cker.org
SSL Certificates: web.h4cker.org
SSL Certificates: websploit.h4cker.org
SSL Certificates: bootcamp.h4cker.org
Yahoo: websploit.h4cker.org
```

Exercise 2.4: Buscador

This exercise is to be completed in your own (after class).

Buscador is a Linux Virtual Machine that is pre-configured for online investigators. It was developed by David Westcott and Michael Bazzell.

1. Download Buscador from: https://inteltechniques.com/buscador/
2. Pick a safe victim (i.e., your own name or your own network).
3. See what you can find out about yourself or your network using tools like Creepy, Shodan, Metagoofil.
4. If you have a website, use the HTTrack Cloner tool to find out how easily it is to clone a site that you can then use to perform social engineering attacks.

Active Reconnaissance

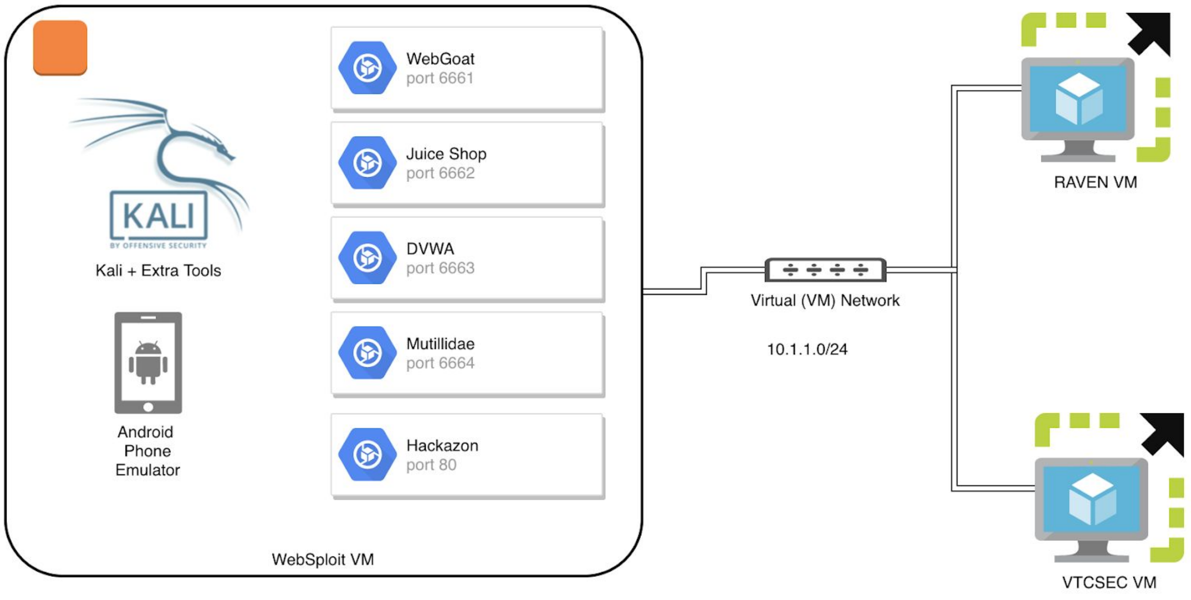
Tip: You can always review the following resources to obtain more information about Active Recon:

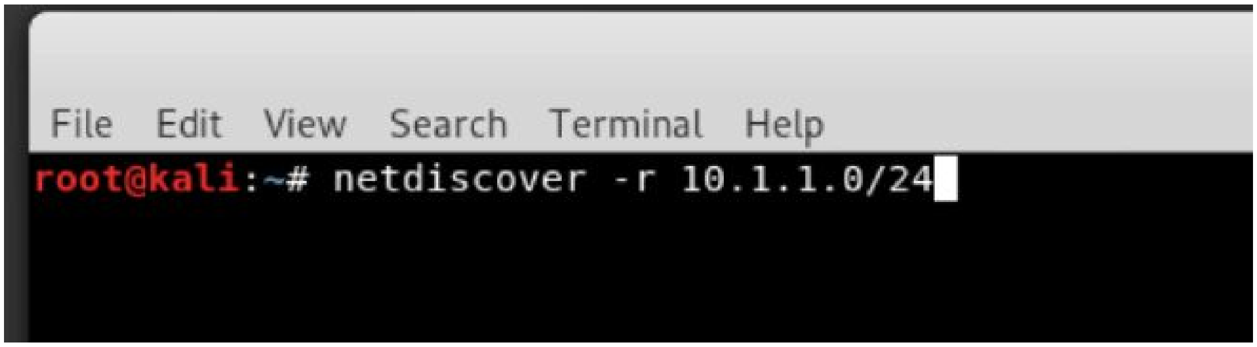
- https://h4cker.org/go/info_gathering
- https://h4cker.org/go/active_recon

Exercise 3.1: Netdiscover

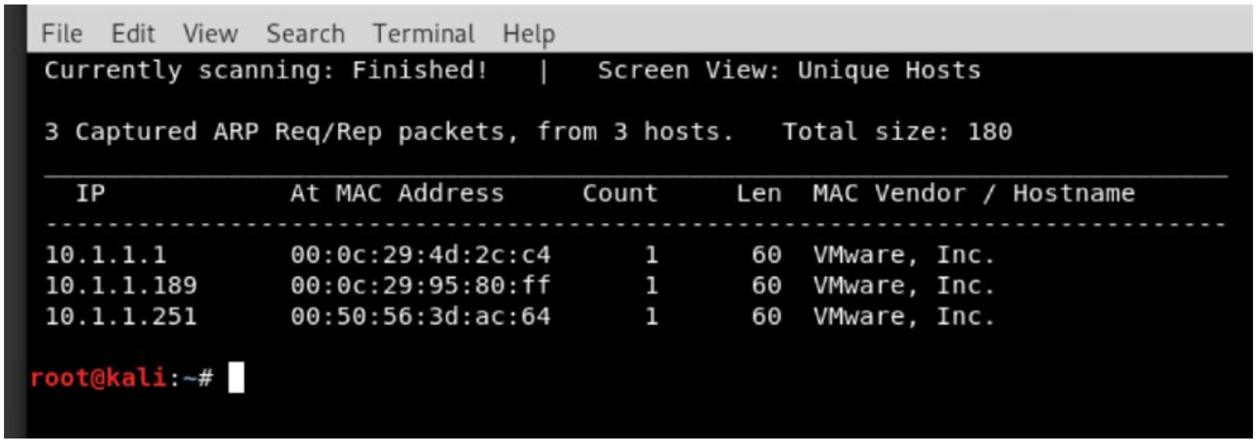
Use Netdiscover to “discover” your victim VMs. In my case, the victim VMs are in the 10.1.1.0/24 network, as illustrated in the following Figure.

Note: Your network address will depend on your Virtual Box or VMWare configuration.





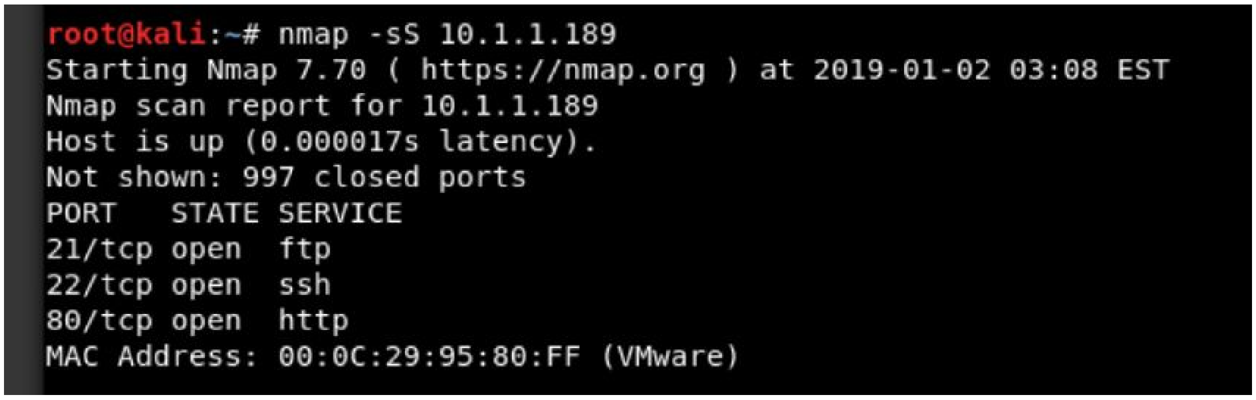
```
root@kali:~# netdiscover -r 10.1.1.0/24
```



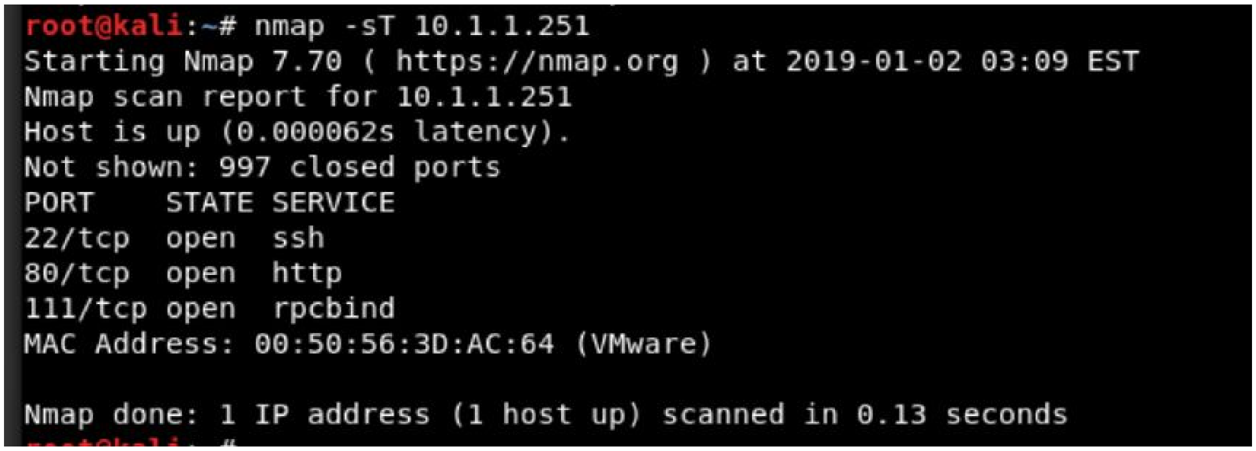
```
 Currently scanning: Finished!   |   Screen View: Unique Hosts

 3 Captured ARP Req/Rep packets, from 3 hosts.   Total size: 180
 _____________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 -----------------------------------------------------------------------------
 10.1.1.1        00:0c:29:4d:2c:c4      1      60  VMware, Inc.
 10.1.1.189      00:0c:29:95:80:ff      1      60  VMware, Inc.
 10.1.1.251      00:50:56:3d:ac:64      1      60  VMware, Inc.

root@kali:~#
```

Exercise 3.2: Nmap

1. Launch nmap from your Kali Linux box.
2. Only scan devices that are in your lab! As we discussed in the class, the best way to do this is to build a local lab with virtual machines on a segregated network. Using nmap try to learn the hosts that are active in your network and all the "victims" you can find.
3. Once you find all the active hosts, try to find all the open TCP and UDP ports on those machines. Nmap doesn't scan all ports by default. It limits itself to 1000 or so common ports. Figure out how to overcome this limitation. Scan the two victim machines:



```
root@kali:~# nmap -sS 10.1.1.189
Starting Nmap 7.70 ( https://nmap.org ) at 2019-01-02 03:08 EST
Nmap scan report for 10.1.1.189
Host is up (0.000017s latency).
Not shown: 997 closed ports
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http
MAC Address: 00:0C:29:95:80:FF (VMware)
```

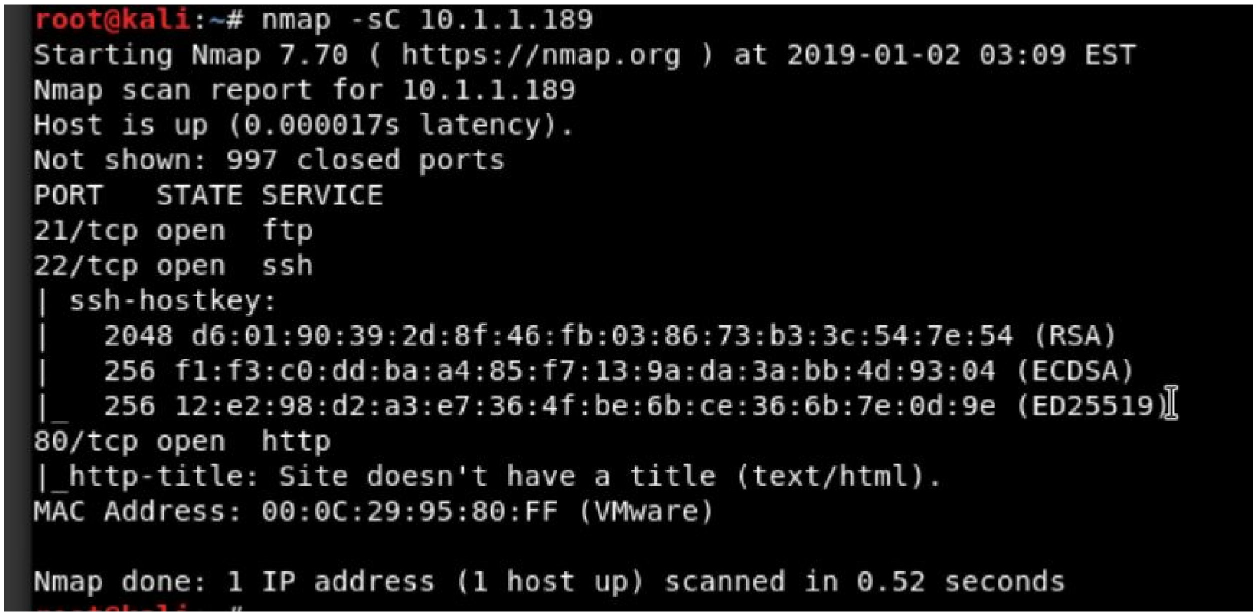


```
root@kali:~# nmap -sT 10.1.1.251
Starting Nmap 7.70 ( https://nmap.org ) at 2019-01-02 03:09 EST
Nmap scan report for 10.1.1.251
Host is up (0.000062s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
111/tcp open  rpcbind
MAC Address: 00:50:56:3D:AC:64 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.13 seconds
```

Notice the ports that are open. You will use them for further attacks. At the end of this course you hack both machines and get root access!!!

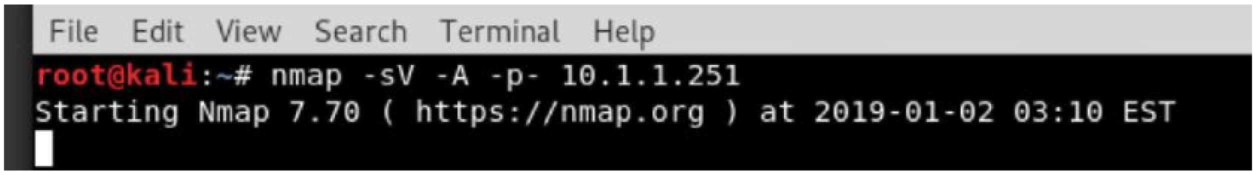
4. Become familiar with the Nmap Scripting Engine (NSE). Locate the scripts. What scripts can you launch against the ports that are open?

Tip: Feel free to use the Nmap cheat sheet at: http://h4cker.org/cheat
In the following example the default scripts are used. However, be creative and explore others.



```
root@kali:~# nmap -sC 10.1.1.189
Starting Nmap 7.70 ( https://nmap.org ) at 2019-01-02 03:09 EST
Nmap scan report for 10.1.1.189
Host is up (0.000017s latency).
Not shown: 997 closed ports
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
| ssh-hostkey:
|   2048 d6:01:90:39:2d:8f:46:fb:03:86:73:b3:3c:54:7e:54 (RSA)
|   256 f1:f3:c0:dd:ba:a4:85:f7:13:9a:da:3a:bb:4d:93:04 (ECDSA)
|_  256 12:e2:98:d2:a3:e7:36:4f:be:6b:ce:36:6b:7e:0d:9e (ED25519)
80/tcp open  http
|_http-title: Site doesn't have a title (text/html).
MAC Address: 00:0C:29:95:80:FF (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.52 seconds
```

5. Add Server Version checks (nmap -sV) to gain more information



```
root@kali:~# nmap -sV -A -p- 10.1.1.251
Starting Nmap 7.70 ( https://nmap.org ) at 2019-01-02 03:10 EST
```

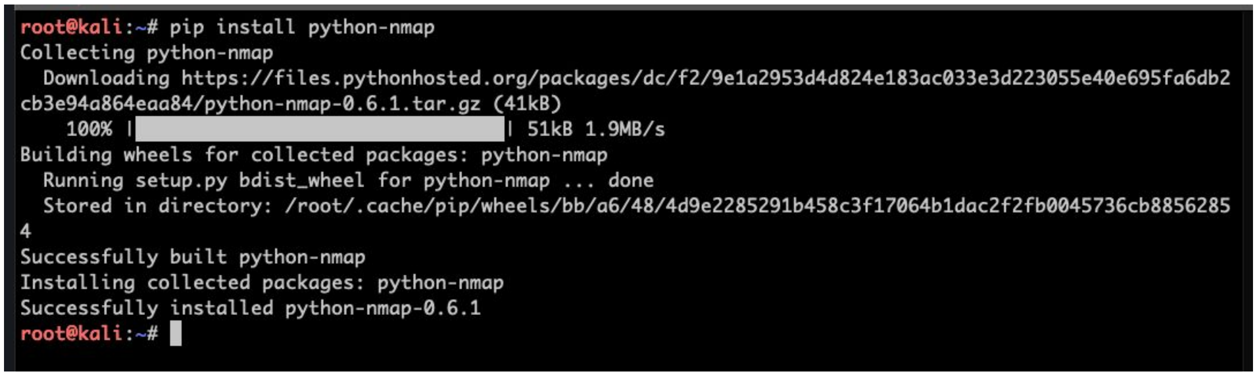
What did you get in your scan results for both servers?

Remember!!! ***Take notes of all the scan results for both victims. You will use them for further attacks. At the end of this course you hack both machines and get root access!!!***

Exercise 3.3: Python and Nmap

python-nmap is a python library which helps in using nmap port scanner and create your own tools. It allows to easily manipulate nmap scan results and is great if you want to automate scanning tasks and reports. It also supports nmap script outputs.

1. Install python-nmap using pip, as shown below:



```
root@kali:~# pip install python-nmap
Collecting python-nmap
  Downloading https://files.pythonhosted.org/packages/dc/f2/9e1a2953d4d824e183ac033e3d223055e40e695fa6db2
cb3e94a864eaa84/python-nmap-0.6.1.tar.gz (41kB)
    100% |████████████████████████████████| 51kB 1.9MB/s
Building wheels for collected packages: python-nmap
  Running setup.py bdist_wheel for python-nmap ... done
  Stored in directory: /root/.cache/pip/wheels/bb/a6/48/4d9e2285291b458c3f17064b1dac2f2fb0045736cb8856285
4
Successfully built python-nmap
Installing collected packages: python-nmap
Successfully installed python-nmap-0.6.1
root@kali:~# 
```

2. Play with the nmap options and try to scan your victim VMs.

```
>>> import nmap
>>> nm = nmap.PortScanner()
>>> nm.scan('10.1.1.251')
```

10.1.1.251 - is one of my victim VMs. You will use your respective IP address space/VM network.

For additional options go to: https://xael.org/pages/python-nmap-en.html

The following are a few examples of things you should be able to do:

```
>>> import nmap
>>> nm = nmap.PortScanner()
>>> nm.scan('10.1.1.251')
{'nmap': {'scanstats': {'uphosts': '1', 'timestr': 'Fri Jan  4 00:26:44
2019', 'downhosts': '0', 'totalhosts': '1', 'elapsed': '6.55'}, 'scaninfo':
{'tcp': {'services':
```

```
'1,3-4,6-7,9,13,17,19-26,30,32-33,37,42-43,49,53,70,79-85,88-90,99-100,106,
109-111,113,119,125,135,139,143-144,146,161,163,179,199,211-212,222,254-256
,259,264,280,301,306,311,340,366,389,406-407,416-417,425,427,443-445,458,46
4-465,481,497,500,512-515,524,541,543-545,548,554-555,563,587,593,616-617,6
25,631,636,646,648,666-668,683,687,691,700,705,711,714,720,722,726,749,765,
777,783,787,800-801,808,843,873,880,888,898,900-903,911-912,981,987,990,992
-993,995,999-1002,1007,1009-1011,1021-1100,1102,1104-1108,1110-1114,1117,11
19,1121-1124,1126,1130-1132,1137-1138,1141,1145,1147-1149,1151-1152,1154,11
63-1166,1169,1174-1175,1183,1185-1187,1192,1198-1199,1201,1213,1216-1218,12
33-1234,1236,1244,1247-1248,1259,1271-1272,1277,1287,1296,1300-1301,1309-13
11,1322,1328,1334,1352,1417,1433-1434,1443,1455,1461,1494,1500-1501,1503,15
21,1524,1533,1556,1580,1583,1594,1600,1641,1658,1666,1687-1688,1700,1717-17
21,1723,1755,1761,1782-1783,1801,1805,1812,1839-1840,1862-1864,1875,1900,19
14,1935,1947,1971-1972,1974,1984,1998-2010,2013,2020-2022,2030,2033-2035,20
38,2040-2043,2045-2049,2065,2068,2099-2100,2103,2105-2107,2111,2119,2121,21
26,2135,2144,2160-2161,2170,2179,2190-2191,2196,2200,2222,2251,2260,2288,23
01,2323,2366,2381-2383,2393-2394,2399,2401,2492,2500,2522,2525,2557,2601-26
02,2604-2605,2607-2608,2638,2701-2702,2710,2717-2718,2725,2800,2809,2811,28
69,2875,2909-2910,2920,2967-2968,2998,3000-3001,3003,3005-3007,3011,3013,30
17,3030-3031,3052,3071,3077,3128,3168,3211,3221,3260-3261,3268-3269,3283,33
00-3301,3306,3322-3325,3333,3351,3367,3369-3372,3389-3390,3404,3476,3493,35
17,3527,3546,3551,3580,3659,3689-3690,3703,3737,3766,3784,3800-3801,3809,38
14,3826-3828,3851,3869,3871,3878,3880,3889,3905,3914,3918,3920,3945,3971,39
86,3995,3998,4000-4006,4045,4111,4125-4126,4129,4224,4242,4279,4321,4343,44
43-4446,4449,4550,4567,4662,4848,4899-4900,4998,5000-5004,5009,5030,5033,50
50-5051,5054,5060-5061,5080,5087,5100-5102,5120,5190,5200,5214,5221-5222,52
25-5226,5269,5280,5298,5357,5405,5414,5431-5432,5440,5500,5510,5544,5550,55
55,5560,5566,5631,5633,5666,5678-5679,5718,5730,5800-5802,5810-5811,5815,58
22,5825,5850,5859,5862,5877,5900-5904,5906-5907,5910-5911,5915,5922,5925,59
50,5952,5959-5963,5987-5989,5998-6007,6009,6025,6059,6100-6101,6106,6112,61
23,6129,6156,6346,6389,6502,6510,6543,6547,6565-6567,6580,6646,6666-6669,66
89,6692,6699,6779,6788-6789,6792,6839,6881,6901,6969,7000-7002,7004,7007,70
19,7025,7070,7100,7103,7106,7200-7201,7402,7435,7443,7496,7512,7625,7627,76
76,7741,7777-7778,7800,7911,7920-7921,7937-7938,7999-8002,8007-8011,8021-80
22,8031,8042,8045,8080-8090,8093,8099-8100,8180-8181,8192-8194,8200,8222,82
54,8290-8292,8300,8333,8383,8400,8402,8443,8500,8600,8649,8651-8652,8654,87
01,8800,8873,8888,8899,8994,9000-9003,9009-9011,9040,9050,9071,9080-9081,90
90-9091,9099-9103,9110-9111,9200,9207,9220,9290,9415,9418,9485,9500,9502-95
03,9535,9575,9593-9595,9618,9666,9876-9878,9898,9900,9917,9929,9943-9944,99
68,9998-10004,10009-10010,10012,10024-10025,10082,10180,10215,10243,10566,1
0616-10617,10621,10626,10628-10629,10778,11110-11111,11967,12000,12174,1226
```

```
5,12345,13456,13722,13782-13783,14000,14238,14441-14442,15000,15002-15004,1
5660,15742,16000-16001,16012,16016,16018,16080,16113,16992-16993,17877,1798
8,18040,18101,18988,19101,19283,19315,19350,19780,19801,19842,20000,20005,2
0031,20221-20222,20828,21571,22939,23502,24444,24800,25734-25735,26214,2700
0,27352-27353,27355-27356,27715,28201,30000,30718,30951,31038,31337,32768-3
2785,33354,33899,34571-34573,35500,38292,40193,40911,41511,42510,44176,4444
2-44443,44501,45100,48080,49152-49161,49163,49165,49167,49175-49176,49400,4
9999-50003,50006,50300,50389,50500,50636,50800,51103,51493,52673,52822,5284
8,52869,54045,54328,55055-55056,55555,55600,56737-56738,57294,57797,58080,6
0020,60443,61532,61900,62078,63331,64623,64680,65000,65129,65389',
'method': 'syn'}}, 'command_line': 'nmap -oX - -sV 10.1.1.251'}, 'scan':
{'10.1.1.251': {'status': {'state': 'up', 'reason': 'arp-response'},
'hostnames': [{'type': '', 'name': ''}], 'vendor': {'00:50:56:3D:AC:64':
'VMware'}, 'addresses': {'mac': '00:50:56:3D:AC:64', 'ipv4': '10.1.1.251'},
'tcp': {80: {'product': 'Apache httpd', 'state': 'open', 'version':
'2.4.10', 'name': 'http', 'conf': '10', 'extrainfo': '(Debian)', 'reason':
'syn-ack', 'cpe': 'cpe:/a:apache:http_server:2.4.10'}, 22: {'product':
'OpenSSH', 'state': 'open', 'version': '6.7p1 Debian 5+deb8u4', 'name':
'ssh', 'conf': '10', 'extrainfo': 'protocol 2.0', 'reason': 'syn-ack',
'cpe': 'cpe:/o:linux:linux_kernel'}, 111: {'product': '', 'state': 'open',
'version': '2-4', 'name': 'rpcbind', 'conf': '10', 'extrainfo': 'RPC
#100000', 'reason': 'syn-ack', 'cpe': ''}}}}}
>>>
```

Additional state and fingerprinting:

```
>>> nm['10.1.1.251'].state
<bound method PortScannerHostDict.state of {'status': {'state': 'up',
'reason': 'arp-response'}, 'hostnames': [{'type': '', 'name': ''}],
'vendor': {'00:50:56:3D:AC:64': 'VMware'}, 'addresses': {'mac':
'00:50:56:3D:AC:64', 'ipv4': '10.1.1.251'}, 'tcp': {80: {'product': 'Apache
httpd', 'state': 'open', 'version': '2.4.10', 'name': 'http', 'conf': '10',
'extrainfo': '(Debian)', 'reason': 'syn-ack', 'cpe':
'cpe:/a:apache:http_server:2.4.10'}, 22: {'product': 'OpenSSH', 'state':
'open', 'version': '6.7p1 Debian 5+deb8u4', 'name': 'ssh', 'conf': '10',
'extrainfo': 'protocol 2.0', 'reason': 'syn-ack', 'cpe':
'cpe:/o:linux:linux_kernel'}, 111: {'product': '', 'state': 'open',
'version': '2-4', 'name': 'rpcbind', 'conf': '10', 'extrainfo': 'RPC
#100000', 'reason': 'syn-ack', 'cpe': ''}}}>
```

Scanning specific ports:

```
>>> nm['10.1.1.251']['tcp'][80]
{'product': 'Apache httpd', 'state': 'open', 'version': '2.4.10', 'name':
'http', 'conf': '10', 'extrainfo': '(Debian)', 'reason': 'syn-ack', 'cpe':
'cpe:/a:apache:http_server:2.4.10'}
```

```
root@kali:~# cat congrats
____ ____ _  _ ____ ____ ____ ___ _  _ _    ____ ___ _ ____ _  _ ____  /
|    |  | |\ | | __ |__/ |__|  |  |  | |    |__|  |  | |  | |\ | [__  /
|___ |__| | \| |__] |  \ |  |  |  |__| |___ |  |  |  | |__| | \| ___] .

                        YOU HAVE COMPLETED DAY 1 !

Tomorrow you will receive another Lab Guide and separate set of slides.


root@kali:~#
```