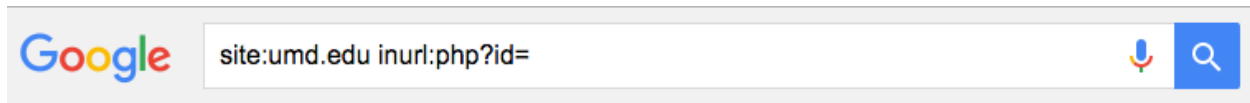


# ENPM685 – Vulnerability Assessment Exercises

Version 3.7 – January 22<sup>nd</sup> 2022

## Google Dorking Exercise (10 minutes)

1. Open a web browser and go to <https://www.google.com>
2. In the search bar try some various “Google Dorks” and try to find something interesting



Example search options:

- **site: domain** (ex. **site: umd.edu**)
- **filetype: doc, pdf, xls, aspx, php**, etc (you can also use **ext** – short for extension here)
- **inurl: something**
- **intext: something**

Examples:

- **site:umd.edu inurl:php?id=**
- **intitle:"netsparker scan report" ext:pdf**
  - Sample result: [http://tdb.gov.in/wp-content/uploads/2017/06/vps49294.vps.ovh.ca\\_80\\_1.pdf](http://tdb.gov.in/wp-content/uploads/2017/06/vps49294.vps.ovh.ca_80_1.pdf)
- **intitle:"Live View / - AXIS" | inurl:view/view.shtml^**
  - Sample result: <http://www.kip.uni-heidelberg.de/view/index.shtml?videos=one>
  - Sample result: <http://camera.buffalotrace.com/view/view.shtml?id=145316&imagePath=/mjpg/video.mjpg&size=1>
- **intitle:"cuckoo sandbox" "failed\_reporting"**
  - Sample result: <http://cuckoosb.com/>
- **inurl:"8080/jmx-console"**
  - <http://www.mobilecard.mx:8080/jmx-console/>
  - <http://www.dmsportal.starhealth.in:8080/jmx-console/>

Exploit DB Google Hacking Database: <https://www.exploit-db.com/google-hacking-database/>

## Shodan Exercise (5 minutes)

Feel free to create an account on Shodan, there is a free option that allows you to unlock more searching and more details on results. If you register with a university email account (something ending in .edu) you may get a free upgrade to a full paid account.

1. Open a web browser and navigate to <https://shodan.io>
2. Start searching for anything you can think of, some examples below (and you can combine multiple search terms into one search)
  - i. Domain names (**umd.edu**)
  - ii. IP addresses
  - iii. Services (**ssh, telnet**)
  - iv. Banners ("**OpenSSH**", "**SSH-1.99**")
  - v. **title:cyberspacekittens**
  - vi. **apache city:"College Park"** (Find Apache servers in College Park)
  - vii. **cisco country:"JP"** (Find cisco equipment in Japan)
  - viii. **"default password"** (Find devices with the default login credentials still set.)

Information on Shodan's search query options: <https://help.shodan.io/the-basics/search-query-fundamentals>

## nmap Exercises – Introduction

1. Many of the nmap port scans we want to do need root privileges so let's create a root shell with **sudo su** and enter your password when asked.
2. In your Kali VM open a terminal and type **nmap ubuntu.ip**

```
(kts@kali) - [~]
$ sudo su
[sudo] password for kts:
(kts@kali) - [~/home/kts]
# nmap 172.16.0.208
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-21 22:21 EST
Nmap scan report for 172.16.0.208
Host is up (0.00034s latency).
Not shown: 993 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
8000/tcp   open  http-alt
8080/tcp   open  http-proxy
8089/tcp   open  unknown
9000/tcp   open  cslistener
MAC Address: 00:0C:29:82:84:CB (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.24 seconds
```

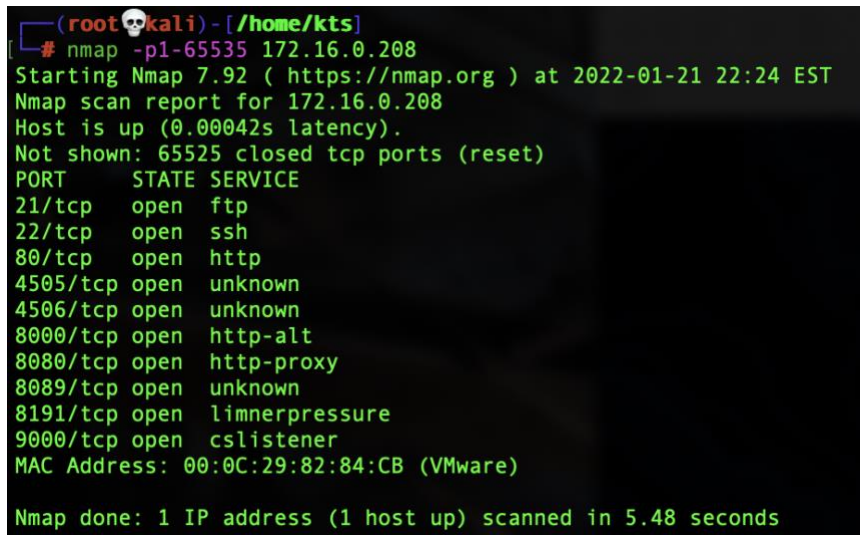
From this we can see that the Ubuntu system appears to be running an FTP server (port 21), SSH server (port 22), Web server (port 80), and some additional ports are open as well – 8000, 8080, 8089, 9000. These are non-standard but maybe with some additional scanning we can determine what they are.

When you don't specify the port range(s) to scan for nmap will default to a list of the 1,000 most common ports.

3. Use the **-p** command line argument to specify specific ports to scan. You have tons of options here, try a few from below and make up some of your own. Some typical ports to quickly look for systems online are: TCP 21, 22, 53, 80, 135, 139, 443, 445, 1433, 3306, 3389

Some examples:

- **nmap -p 22 172.28.128.5**
- **nmap -p 22,80,443,8000 172.28.128.5**
- **nmap -p 1-65535 172.28.128.5**



```
(root@kali) - [ /home/kts ]
[ # nmap -p1-65535 172.16.0.208
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-21 22:24 EST
Nmap scan report for 172.16.0.208
Host is up (0.00042s latency).
Not shown: 65525 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
4505/tcp  open  unknown
4506/tcp  open  unknown
8000/tcp  open  http-alt
8080/tcp  open  http-proxy
8089/tcp  open  unknown
8191/tcp  open  limerpressure
9000/tcp  open  cslistener
MAC Address: 00:0C:29:82:84:CB (VMware)

Nmap done: 1 IP address (1 host up) scanned in 5.48 seconds
```

4. nmap will let you specify the target systems by a number of different options. Some examples:
- **nmap -p 22 172.28.128.5**
  - **nmap -p 22 172.28.128.0/24**
  - **nmap -p 22 172.28.128.1-10**
  - **nmap -p 22 scanme.nmap.org** (Note: a real system you can scan)
  - **nmap -p 22 -iL filename** (Note: where *filename* is a file that contains a list of IP addresses/subnets/DNS names/etc like the examples listed above)

```

(root@kali) - [/home/kts]
# nmap 172.16.0.200-210
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-21 22:28 EST
Nmap scan report for 172.16.0.207
Host is up (0.00091s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
8000/tcp  open  http-alt
8080/tcp  open  http-proxy
8089/tcp  open  unknown
MAC Address: 00:0C:29:81:4E:8C (VMware)

Nmap scan report for 172.16.0.208
Host is up (0.00080s latency).
Not shown: 993 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
8000/tcp  open  http-alt
8080/tcp  open  http-proxy
8089/tcp  open  unknown
9000/tcp  open  cslistener
MAC Address: 00:0C:29:82:84:CB (VMware)

Nmap scan report for 172.16.0.206
Host is up (0.0000030s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh

Nmap done: 11 IP addresses (3 hosts up) scanned in 1.66 seconds

```

5. Try the OS Fingerprinting module by using the **-O** command line argument.  
(Ex: **nmap -O ubuntu.ip**)

```

(root@kali) - [/home/kts]
# nmap -O 172.16.0.208
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-21 22:30 EST
Nmap scan report for 172.16.0.208
Host is up (0.00068s latency).
Not shown: 993 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
8000/tcp  open  http-alt
8080/tcp  open  http-proxy
8089/tcp  open  unknown
9000/tcp  open  cslistener
MAC Address: 00:0C:29:82:84:CB (VMware)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/sub
mit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.63 seconds

```

You see at the bottom of the output from nmap that nmap believe the system you just scanned is a Linux system running a kernel version 4.15 – 5.6.

6. Try the Service Versioning module by using the **-sV** command line argument.  
(Ex: **nmap -sV -p1-65535 ubuntu.ip**)

```
(root@kali)~# nmap -sV -p1-65535 172.16.0.208
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-21 22:37 EST
Nmap scan report for 172.16.0.208
Host is up (0.00069s latency).
Not shown: 65525 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 3.0.3
22/tcp    open  ssh          OpenSSH 8.2p1 Ubuntu 4ubuntu0.4 (Ubuntu Linux; proto
col 2.0)
80/tcp    open  http         Apache httpd 2.4.41 ((Ubuntu))
4505/tcp  open  zmtmp       ZeroMQ ZMTP 2.0
4506/tcp  open  zmtmp       ZeroMQ ZMTP 2.0
8080/tcp  open  ssl/http    CherryPy wsgiserver
8080/tcp  open  http        Jetty 9.4.43.v20210629
8089/tcp  open  ssl/http    Splunkd httpd
8191/tcp  open  limnerpressure?
9000/tcp  open  http        Splunkd httpd
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port8191-TCP:V=7.92%I=7%D=1/21%Time=61EB7C17%P=x86_64-pc-linux-gnu%r(Ge
SF:tRequest,A9,"HTTP/1.0\x20200\x20OK\r\nConnection:\x20close\r\nContent-
SF:Type:\x20text/plain\r\nContent-Length:\x2085\r\n\r\nIt\x20looks\x20like
SF:\x20you\x20are\x20trying\x20to\x20access\x20MongoDB\x20over\x20HTTP\x20
SF:on\x20the\x20native\x20driver\x20port\.\r\n")%r(FourOhFourRequest,A9,"H
SF:TTP/1.0\x20200\x20OK\r\nConnection:\x20close\r\nContent-Type:\x20text/
SF:plain\r\nContent-Length:\x2085\r\n\r\nIt\x20looks\x20like\x20you\x20are
SF:\x20trying\x20to\x20access\x20MongoDB\x20over\x20HTTP\x20on\x20the\x20n
SF:ative\x20driver\x20port\.\r\n");
MAC Address: 00:0C:29:82:84:CB (VMware)
Service Info: 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 54.26 seconds
```

You can see that nmap detected the versions of the FTP server, SSH server, Web server, detected that ports 8089 and 9000 are related to a Splunk install, and we also learned some more about ports 4505, 4506, 8000, and 8080. Some further analysis may help us determine more.

7. The **-A** option pretty much stands for “all”. It turns on OS fingerprinting, service versioning, NSE scripts, and even a traceroute. Example: **nmap -A 172.28.128.5**



```

(root@kali)-[/home/kts]
# nmap -A 172.16.0.208
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-21 22:47 EST
Nmap scan report for 172.16.0.208
Host is up (0.00057s latency).
Not shown: 993 closed tcp ports (reset)
PORT      STATE SERVICE  VERSION
21/tcp    open  ftp      vsftpd 3.0.3
| ftp-syst:
|   STAT:
|   FTP server status:
|     Connected to ::ffff:172.16.0.206
|     Logged in as ftp
|     TYPE: ASCII
|     No session bandwidth limit
|     Session timeout in seconds is 300
|     Control connection is plain text
|     Data connections will be plain text
|     At session startup, client count was 1
|     vsFTPD 3.0.3 - secure, fast, stable
|_End of status
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| drwxr-xr-x   2 0          0          4096 Jan 17 03:08 backup
|_ -rw-r--r--   1 0          0          27 Jan 17 03:08 secret.txt
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 ad:3e:d8:48:b8:e0:29:74:a5:02:85:f7:a9:14:d7:6b (RSA)
|   256 e0:fc:59:99:38:ae:b6:32:8e:c5:dc:56:c3:f8:6d:f8 (ECDSA)
|_  256 57:d7:4a:47:39:42:38:3a:98:41:11:d9:88:7f:18:f2 (ED25519)
80/tcp    open  http     Apache httpd 2.4.41 ((Ubuntu))
|_http-title: ENPM685 Dojo
|_http-server-header: Apache/2.4.41 (Ubuntu)
| http-robots.txt: 1 disallowed entry
|_ /phpmyadmin
8000/tcp  open  ssl/http CherryPy wsgiserver
| ssl-cert: Subject: commonName=localhost/organizationName=SaltStack/stateOrProvinceName=Utah/countryName=US
| Not valid before: 2022-01-08T04:14:28
|_Not valid after:  2023-01-08T04:14:28
|_http-server-header: CherryPy/8.9.1
(output snippet)

```

8. While you can always copy and paste output to a text file (or use **> output.txt** at the end of the command to put it into a file called 'output.txt') the multiline results can be a pain. nmap has robust reporting outputs in a variety of formats (you have been warned about -oS) the most useful might be **-oG** for Greppable output.

Example: **nmap -p 22 -oG output.txt 172.16.0.0/24**

The results are easy to review with **grep**, **sed**, **awk**, or any of your favorite scripting languages. (ex. **grep "22/open" output.txt**)

```

(root@kali)-[/home/kts]
# grep "22/open" output.txt
Host: 172.16.0.207 () Ports: 22/open/tcp//ssh///
Host: 172.16.0.208 () Ports: 22/open/tcp//ssh///
Host: 172.16.0.206 () Ports: 22/open/tcp//ssh///

```

## nmap Script Engine Examples – Advanced

NSE extends nmap to help you quickly find out more information about a host. NSE scripts are saved with a .nse file extension. In Kali the default location is **/usr/share/nmap/scripts**. You can use wildcards (\* and ?) when selecting which NSE scripts to run and you can use “and not” to exclude specific scripts (for example **ssh-brute**.)

Run a number of common HTTP scripts to learn more information about the server:

**nmap -p 80 --script "http-headers,http-title,http-robots\*,http-server-header,http-useragent-tester" ubuntu.ip**

```
(root@kali)~[/home/kts]
# nmap -p 80 --script "http-headers,http-title,http-robots*,http-server-header,http-useragent-tester" 172.16.0.208
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-21 23:18 EST
Nmap scan report for 172.16.0.208
Host is up (0.00045s latency).

PORT      STATE SERVICE
80/tcp    open  http
|_http-title: ENPM685 Dojo
|_http-headers:
|   Date: Sat, 22 Jan 2022 04:18:20 GMT
|   Server: Apache/2.4.41 (Ubuntu)
|   Connection: close
|   Content-Type: text/html; charset=UTF-8
|_ (Request type: HEAD)
|_http-robots.txt: 1 disallowed entry
|_/phpmyadmin
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-useragent-tester:
|   Status for browser useragent: 200
|   Allowed User Agents:
|     Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)
|     libwww
|     lwp-trivial
|     libcurl-agent/1.0
|     PHP/
|     Python-urllib/2.5
|     GT:WWW
|     Snoopy
|     MFC_Tear_Sample
|     HTTP::Lite
|     PHPCrawl
|     URI::Fetch
|     Zend_Http_Client
|     http_client
|     PECL::HTTP
|     Wget/1.13.4 (linux-gnu)
|     WWW-Mechanize/1.34
|_ MAC Address: 00:0C:29:82:84:CB (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.31 seconds
```

Enumerate all SSH scripts other than **ssh-brute** against a host: **nmap -p 22 --script "ssh\*" and not ssh-brute ubuntu.ip**

```
PORT      STATE SERVICE
22/tcp    open  ssh
| ssh-hostkey:
|   3072 ad:3e:d8:48:b8:e0:29:74:a5:02:85:f7:a9:14:d7:6b (RSA)
|   256  e0:fc:59:99:38:ae:b6:32:8e:c5:dc:56:c3:f8:6d:f8 (ECDSA)
|_  256  57:d7:4a:47:39:42:38:3a:98:41:11:d9:88:7f:18:f2 (ED25519)
|_ ssh-run: Failed to specify credentials and command to run.
| ssh2-enum-algos:
|   kex_algorithms: (9)
|     curve25519-sha256
|     curve25519-sha256@libssh.org
|     ecdh-sha2-nistp256
|     ecdh-sha2-nistp384
|     ecdh-sha2-nistp521
|     diffie-hellman-group-exchange-sha256
|     diffie-hellman-group16-sha512
|     diffie-hellman-group18-sha512
|     diffie-hellman-group14-sha256
|   server_host_key_algorithms: (5)
|     rsa-sha2-512
|     rsa-sha2-256
|     ssh-rsa
|     ecdsa-sha2-nistp256
|     ssh-ed25519
|   encryption_algorithms: (6)
|     chacha20-poly1305@openssh.com
|     aes128-ctr
|     aes192-ctr
|     aes256-ctr
|     aes128-gcm@openssh.com
|     aes256-gcm@openssh.com
|   mac_algorithms: (10)
|     umac-64-etm@openssh.com
|     umac-128-etm@openssh.com
|     hmac-sha2-256-etm@openssh.com
|     hmac-sha2-512-etm@openssh.com
|     hmac-sha1-etm@openssh.com
|     umac-64@openssh.com
|     umac-128@openssh.com
|     hmac-sha2-256
|     hmac-sha2-512
|     hmac-sha1
|   compression_algorithms: (2)
|     none
|_    zlib@openssh.com
|_ ssh-auth-methods:
|   Supported authentication methods:
|     publickey
|_    password
```

(Output trimmed. You'll see a lot of SSH login attempts)



Enumerate SSL ciphers a host allows:

```
nmap -p443 --script ssl-enum-ciphers www.umd.edu
```

Sample output:

```
(root@kali)~[/home/kts]
# nmap -p443 --script ssl-enum-ciphers www.umd.edu
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-21 23:28 EST
Nmap scan report for www.umd.edu (13.226.31.64)
Host is up (0.0022s latency).
Other addresses for www.umd.edu (not scanned): 13.226.31.45 13.226.31.39 13.226.31.125
rDNS record for 13.226.31.64: server-13-226-31-64.ewr53.r.cloudfront.net

PORT      STATE SERVICE
443/tcp   open  https
| ssl-enum-ciphers:
|   TLSv1.2:
|     ciphers:
|       TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (ecd_h_x25519) - A
|       TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (ecd_h_x25519) - A
|       TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (ecd_h_x25519) - A
|       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (ecd_h_x25519) - A
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (ecd_h_x25519) - A
|     compressors:
|       NULL
|     cipher preference: server
|   TLSv1.3:
|     ciphers:
|       TLS_AKE_WITH_AES_128_GCM_SHA256 (ecd_h_x25519) - A
|       TLS_AKE_WITH_AES_256_GCM_SHA384 (ecd_h_x25519) - A
|       TLS_AKE_WITH_CHACHA20_POLY1305_SHA256 (ecd_h_x25519) - A
|     cipher preference: server
|_  least strength: A

Nmap done: 1 IP address (1 host up) scanned in 1.45 seconds
```

From our earlier port scans we see that port 8000 is running a webserver (**CherryPy wsgiserver**) let's use nmap to help us learn more about what may be running on port 8000.

```
nmap -p 8000 --script "http-headers,http-title,http-robots*,http-server-header,http-useragent-tester" ubuntu.ip
```

```
(root@kali)~[/home/kts]
# nmap -p 8000 --script "http-headers,http-title,http-robots*,http-server-header,http-useragent-tester" 172.16.0.208
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-22 00:37 EST
Nmap scan report for 172.16.0.208
Host is up (0.00053s latency).

PORT      STATE SERVICE
8000/tcp   open  http-alt
|_ http-server-header: CherryPy/8.9.1
|_ http-title: Site doesn't have a title (application/json).
| http-headers:
|   Content-Type: application/json
|   Server: CherryPy/8.9.1
|   Date: Sat, 22 Jan 2022 05:37:04 GMT
|   Allow: GET, HEAD, POST
|   Access-Control-Allow-Origin: *
|   Access-Control-Expose-Headers: GET, POST
|   Access-Control-Allow-Credentials: true
|   Vary: Accept-Encoding
|   Content-Length: 146
|   Connection: close
|_ (Request type: HEAD)
MAC Address: 00:0C:29:82:84:CB (VMware)
```

Let's do some SSL checks too - `nmap -p 8000 --script "ssl*" 172.16.0.208`

```
PORT      STATE SERVICE
8000/tcp  open  http-alt
|_ssl-date: TLS randomness does not represent time
|_ssl-enum-ciphers:
|   TLSv1.2:
|     ciphers:
|       TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (ecdh_x25519) - A
|       TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (ecdh_x25519) - A
|       TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (ecdh_x25519) - A
|       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (ecdh_x25519) - A
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (ecdh_x25519) - A
|       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (ecdh_x25519) - A
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (ecdh_x25519) - A
|       TLS_RSA_WITH_AES_256_GCM_SHA384 (rsa 2048) - A
|       TLS_RSA_WITH_AES_128_GCM_SHA256 (rsa 2048) - A
|       TLS_RSA_WITH_AES_256_CBC_SHA256 (rsa 2048) - A
|       TLS_RSA_WITH_AES_128_CBC_SHA256 (rsa 2048) - A
|       TLS_RSA_WITH_AES_256_CBC_SHA (rsa 2048) - A
|       TLS_RSA_WITH_AES_128_CBC_SHA (rsa 2048) - A
|     compressors:
|       NULL
|     cipher preference: server
|   TLSv1.3:
|     ciphers:
|       TLS_AKE_WITH_AES_256_GCM_SHA384 (ecdh_x25519) - A
|       TLS_AKE_WITH_CHACHA20_POLY1305_SHA256 (ecdh_x25519) - A
|       TLS_AKE_WITH_AES_128_GCM_SHA256 (ecdh_x25519) - A
|     cipher preference: server
|_ least strength: A
|_ssl-cert: Subject: commonName=localhost/organizationName=SaltStack/stateOrProvinceName=Utah/countryName=US
|_Issuer: commonName=localhost/organizationName=SaltStack/stateOrProvinceName=Utah/countryName=US
|_Public Key type: rsa
|_Public Key bits: 2048
|_Signature Algorithm: sha256WithRSAEncryption
|_Not valid before: 2022-01-08T04:14:28
|_Not valid after: 2023-01-08T04:14:28
|_MD5: a621 7487 8f42 a2b8 3b0d d7d2 7fd2 6276
|_SHA-1: ea1c dc2b 026c 3867 ae8b 7e7e 83de 8790 b86f d802
```

Highlighted in the SSL certificate is “SaltStack” which is interesting. If you are unfamiliar with SaltStack some Google-fu can help you discover that SaltStack is a tool for automation, remote task execution, and configuration management. It typically runs on ports 4505 and 4506 (which are also open on the Ubuntu VM) and it often has a web-based API that runs on port 8000 with CherryPy as the Python-base web framework that the API utilizes.

More on this:

- [https://salt-sproxy.readthedocs.io/en/latest/salt\\_api.html](https://salt-sproxy.readthedocs.io/en/latest/salt_api.html)
- <https://www.speedguide.net/port.php?port=4505>
- <https://www.speedguide.net/port.php?port=4506>

We will come back to this information when we move into the penetration testing section.

## Nikto Exercises (5 minutes)

1. On your Kali VM in a Terminal type “**nikto -h ubuntu.ip**”
2. Let Nikto run and then review the results.

```
(root@kali)~# nikto -h 172.16.0.208
- Nikto v2.1.6

+-----+
+ Target IP:      172.16.0.208
+ Target Hostname: 172.16.0.208
+ Target Port:    80
+ Start Time:     2022-01-21 23:29:46 (GMT-5)
+-----+

+ Server: Apache/2.4.41 (Ubuntu)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to
protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to rende
r the content of the site in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ "robots.txt" contains 1 entry which should be manually viewed.
+ Web Server returns a valid response with junk HTTP methods, this may cause false posit
ives.
+ OSVDB-3268: /logs/: Directory indexing found.
+ OSVDB-3092: /logs/: This might be interesting...
+ /info.php: Output from the phpinfo() function was found.
+ OSVDB-3233: /info.php: PHP is installed, and a test script which runs phpinfo() was fo
und. This gives a lot of system information.
+ OSVDB-5292: /info.php?file=http://cirt.net/rfiinc.txt?: RFI from RSNAKE's list (http:/
/ha.ckers.org/weird/rfi-locations.dat) or from http://osvdb.org/
+ 8070 requests: 0 error(s) and 10 item(s) reported on remote host
+ End Time:       2022-01-21 23:30:33 (GMT-5) (47 seconds)
+-----+

+ 1 host(s) tested
```

If asked to submit the HTTP Server banner information select **No**.

3. What interesting things do you find?

Additional command line arguments you may find useful:

<b>-Format txt</b>	Output data in standard text format (other options – csv, htm, xml, nbe)
<b>-o results.txt</b>	Save the results to the file named results.txt
<b>-update</b>	Update Nikto's plugins

## WPScan Exercise (5 minutes)

1. On Kali in a Terminal type “**wpscan --update**” to update wpscan’s database of vulnerabilities
2. Try wpscan against a site you know that runs Wordpress
  - a. ex: **wpscan --url http://msmc.umd.edu** (note you may need to add **--random-user-agent**)
3. After the scan completes review the results, you’ll find lots of interesting things. The lesson? Don’t run WordPress. Or Drupal. (Or if you have to always be patching.)

```
[+] URL: http://msmc.umd.edu/ [52.22.105.195]
[+] Effective URL: https://msmc.umd.edu:443/
[+] Started: Fri Jan 21 23:33:57 2022

Interesting Finding(s):

[+] Headers
| Interesting Entries:
|   - server: Apache
|   - x-powered-by: PHP/7.4.13
| Found By: Headers (Passive Detection)
| Confidence: 100%

[+] XML-RPC seems to be enabled: https://msmc.umd.edu/xmlrpc.php
| Found By: Link Tag (Passive Detection)
| Confidence: 30%
| References:
|   - http://codex.wordpress.org/XML-RPC_Pingback_API
|   - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
|   - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
|   - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
|   - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/

[+] WordPress version 4.7.3 identified (Insecure, released on 2017-03-06).
| Found By: Rss Generator (Passive Detection)
|   - https://msmc.umd.edu/index.php/feed/, <generator>https://wordpress.org/?v=4.7.3</generator>
|   - https://msmc.umd.edu/index.php/comments/feed/, <generator>https://wordpress.org/?v=4.7.3</generator>

[+] WordPress theme in use: hemingway-rewritten-wpcom
| Location: http://msmc.umd.edu/wp-content/themes/hemingway-rewritten-wpcom/
| Style URL: https://msmc.umd.edu/wp-content/themes/hemingway-rewritten-wpcom/style.css?ver=4.7.3
| Style Name: Hemingway Rewritten - WordPress.com
| Style URI: https://wordpress.com/themes/hemingway-rewritten/
| Description: Hemingway Rewritten is a classic blog theme with a parallax-scrolling header effect and a minimal, e...
| Author: Anders Noren
| Author URI: http://www.andersnoren.se
| Found By: Css Style In Homepage (Passive Detection)
| Confirmed By: Css Style In 404 Page (Passive Detection)
| Version: 1.1.4 (80% confidence)
| Found By: Style (Passive Detection)
|   - https://msmc.umd.edu/wp-content/themes/hemingway-rewritten-wpcom/style.css?ver=4.7,
```

(output trimmed)

## Nessus Exercises (30 minutes)

**Note:** If you did not install Nessus for part of Homework #1 now is the time to install Nessus. See the Homework #1 document for the steps to complete this task.

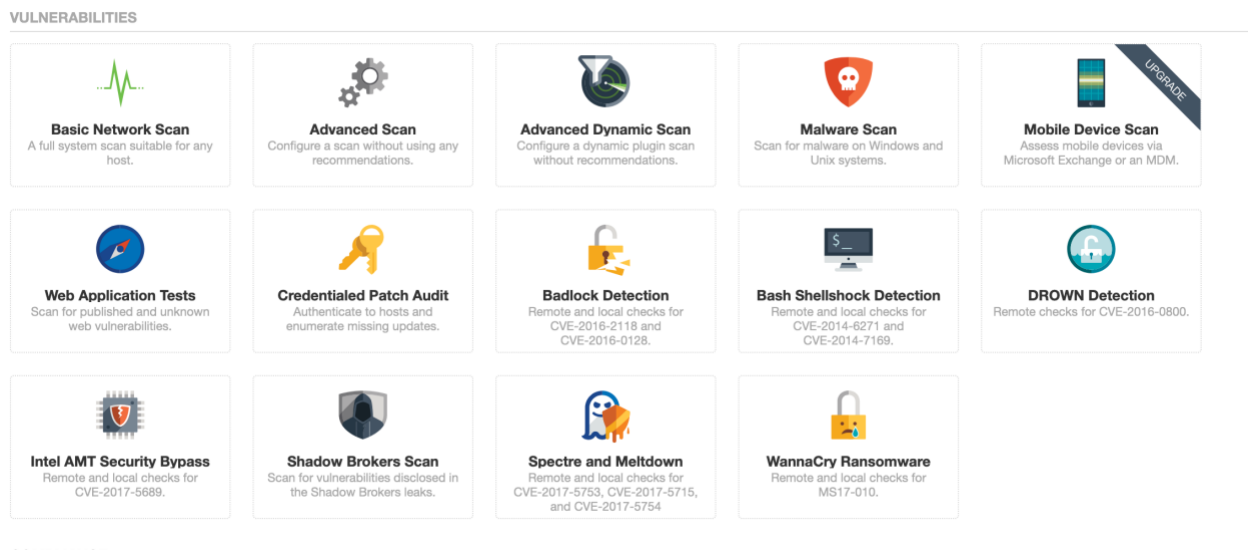
Note: if you notice that the web UI for Nessus does not load and checking with `ps/netstat` shows that Nessus is not running you can start it with `sudo /bin/systemctl start nessusd.service`

### Scanning with Nessus

1. In the main Nessus web UI screen click the “+ New Scan” button at the top right



2. You'll see a large number of options we can scan with. Click **Advanced Scan** at the top left.



3. Fill out the information needed in the **Settings** tab: give your scan a name, a description if you wish, and the target – in this case our Ubuntu VM

The screenshot shows the 'Settings' tab with the 'BASIC' section expanded. The 'Name' field contains 'Ubuntu VM', 'Description' is empty, 'Folder' is set to 'My Scans', and 'Targets' contains '172.16.0.193'. At the bottom, there are 'Save' and 'Cancel' buttons.

Feel free to view the other options on the “**Settings**” tab, the defaults are fine for our needs. (Under “**Discovery**” -> “**Port Scanning**” it only scans a list of “**default**” (common) ports. For most needs this is fine but if you are only scanning a few systems scanning all ports (**1-65535**) can sometimes turn up interesting information.

4. Scroll down and click “**Save**” at the bottom.
5. This will take you back to the My Scans page. Click the “Launch” button that looks like a Play/right arrow button on the far right of your scan.

The screenshot shows the 'My Scans' page with a table containing one scan entry. The 'Launch' button is visible on the right side of the row.

Name	Schedule	Last Modified	Launch
ENPM685 Class Scan	On Demand	N/A	▶

6. The scan will launch and take some time to run. Feel free to grab a drink, check your email, plot for world domination, etc.

The screenshot shows the 'My Scans' page with the scan entry 'ENPM685 Class Scan' now showing a spinning circle icon, indicating the scan is in progress.

Name	Schedule	Last Modified	
ENPM685 Class Scan	On Demand	Today at 10:15 PM	⏸

When completed you’ll see the spinning circle change to a check box.

The screenshot shows the 'My Scans' page with the scan entry 'ENPM685 Class Scan' now showing a checkmark icon, indicating the scan is completed.

Name	Schedule	Completed	Last Modified	
ENPM685 Class Scan	On Demand	✓	Today at 10:17 PM	▶

7. Click anywhere in the row that shows your scan (“**ENPM685 Class Scan**” in my example) to be taken to the results.



## Sample Scan

[Back to My Scans](#)

Configure

Audit Trail

Launch ▼

Report

Export ▼

Hosts 1

Vulnerabilities 30

VPR Top Threats 0

History 1

Filter ▼

Search Hosts



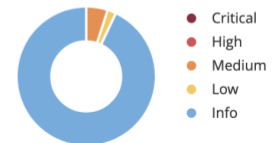
1 Host

<input type="checkbox"/>	Host	Vulnerabilities ▼
<input type="checkbox"/>	172.16.0.208	<div><div>6</div><div>77</div></div>

### Scan Details

Policy: Advanced Scan  
Status: Completed  
Severity Base: CVSS v3.0  
Scanner: Local Scanner  
Start: January 21 at 11:46 PM  
End: Today at 12:03 AM  
Elapsed: 17 minutes

### Vulnerabilities



8. Explore the results. (You can either click on the Host IP or the Vulnerabilities tab to get more details about the vulnerabilities on the host.)

Hosts1

Vulnerabilities30

VPR Top Threats3

History1

Filter

Search Vulnerabilities

30 Vulnerabilities

<input type="checkbox"/>	Sev	Score	Name	Family	Count		
<input type="checkbox"/>	MIXED	...	7 SSL (Multiple Issues)	General	19	🕒	✎
<input type="checkbox"/>	MIXED	...	2 TLS (Multiple Issues)	General	4	🕒	✎
<input type="checkbox"/>	INFO	...	4 HTTP (Multiple Issues)	Web Servers	9	🕒	✎
<input type="checkbox"/>	INFO	...	2 TLS (Multiple Issues)	Service detection	4	🕒	✎
<input type="checkbox"/>	INFO	...	2 Splunk (Multiple Issues)	Web Servers	3	🕒	✎
<input type="checkbox"/>	INFO	...	2 SSH (Multiple Issues)	Misc.	2	🕒	✎
<input type="checkbox"/>	INFO	...	2 SSH (Multiple Issues)	Service detection	2	🕒	✎
<input type="checkbox"/>	INFO		Service Detection	Service detection	11	🕒	✎

9. Click the “SSL (Multiple Issues)” listing at the top to get more details.

Hosts1
















Vulnerabilities30

VPR Top Threats

History1

Search Vulnerabilities

7 Vulnerabilities

<input type="checkbox"/>	Sev ▾	Score ▾	Name ▴	Family ▴	Count ▾	
<input type="checkbox"/>	MEDIUM	6.5	SSL Certificate Cannot Be Tru...	General	3	 
<input type="checkbox"/>	MEDIUM	6.4 *	SSL Self-Signed Certificate	General	3	 
<input type="checkbox"/>	INFO		SSL Certificate Information	General	3	 
<input type="checkbox"/>	INFO		SSL Cipher Block Chaining Cl...	General	3	 
<input type="checkbox"/>	INFO		SSL Cipher Suites Supported	General	3	 
<input type="checkbox"/>	INFO		SSL Perfect Forward Secrecy ...	General	3	 
<input type="checkbox"/>	INFO		SSL Compression Methods S...	General	1	 

10. You can click on the vulnerabilities to get more details. I clicked the top one for this example.

MEDIUM

SSL Certificate Cannot Be Trusted

**Description**

The server's X.509 certificate cannot be trusted. This situation can occur in three different ways, in which the chain of trust can be broken, as stated below :

- First, the top of the certificate chain sent by the server might not be descended from a known public certificate authority. This can occur either when the top of the chain is an unrecognized, self-signed certificate, or when intermediate certificates are missing that would connect the top of the certificate chain to a known public certificate authority.
- Second, the certificate chain may contain a certificate that is not valid at the time of the scan. This can occur either when the scan occurs before one of the certificate's 'notBefore' dates, or after one of the certificate's 'notAfter' dates.
- Third, the certificate chain may contain a signature that either didn't match the certificate's information or could not be verified. Bad signatures can be fixed by getting the certificate with the bad signature to be re-signed by its issuer. Signatures that could not be verified are the result of the certificate's issuer using a signing algorithm that Nessus either does not support or does not recognize.

If the remote host is a public host in production, any break in the chain makes it more difficult for users to verify the authenticity and identity of the web server. This could make it easier to carry out man-in-the-middle attacks against the remote host.

**Solution**

Purchase or generate a proper SSL certificate for this service.

**See Also**

<https://www.itu.int/rec/T-REC-X.509/en>  
<https://en.wikipedia.org/wiki/X.509>

**Output**

```
The following certificate was at the top of the certificate
chain sent by the remote host, but it is signed by an unknown
certificate authority :

|-Subject : C=US/ST=Utah/L=Salt Lake City/O=SaltStack/CN=localhost
|-Issuer : C=US/ST=Utah/L=Salt Lake City/O=SaltStack/CN=localhost
```

Port	Hosts
8000 / tcp / www	172.16.0.208

This vulnerability finding is for port 8000 which we first saw when we were port scanning with nmap and it was identified as “**CherryPy wsgiserver**”. Looking at the Subject and Issuer of this SSL certificate we see “**SaltStack**”. If you are unfamiliar with SaltStack some Google-fu can help you discover that SaltStack is a tool for automation, remote task execution, and configuration management. It typically runs on ports 4505 and 4506 (which are also open on the Ubuntu VM) and it often has a web-based API that runs on port 8000 with CherryPy as the Python-base web framework that the API utilizes.

More on this:

- [https://salt-sproxy.readthedocs.io/en/latest/salt\\_api.html](https://salt-sproxy.readthedocs.io/en/latest/salt_api.html)
- <https://www.speedguide.net/port.php?port=4505>
- <https://www.speedguide.net/port.php?port=4506>

We will come back to this information when we move into the penetration testing section.

11. Let’s save a report of this scan. First, we need to get back to the full details of the scan, click “**Back to Vulnerability Group**” and then “**Back to Vulnerabilities**”.
12. Click **Report** at the top right of the UI. This will open up a dialog box and the default settings are good for now, click Generate Report at the bottom left of the dialog box.

13. The report will take a minute or two to run and then will automatically download. Open the report and take a look. At you can see it is a large report. Inside is every issue that was detected along with remediation information and the “proof” of the vulnerability. This is a lot of information to sort through and why if you are into vulnerability scanning or penetration testing you should always provide your customer/management with a

guide on how to process these results. Focus on what they should tackle first based on risk and what the quick wins are. Don't forget to utilize your asset inventory to help determine the risk level.

What should they tackle first? Vulnerabilities that are on critical systems to the business operation and/or house/process sensitive data.

What are quick wins? A (hopefully) simple process that removes a lot of vulnerabilities/risk.

Examples:

- If OpenSSH is running on the system and it's out to date and vulnerable but not in use – turn it off.
- If you have 100+ vulnerabilities because you're running an out-of-date version of PHP and you can update PHP with no impact to any web apps using PHP than updating PHP will quickly resolve a lot of your issues.