COMP 7904
INFORMATION SECURITY: ATTACKS AND DEFENSE

LAB 1 PART 1 WARM-UP

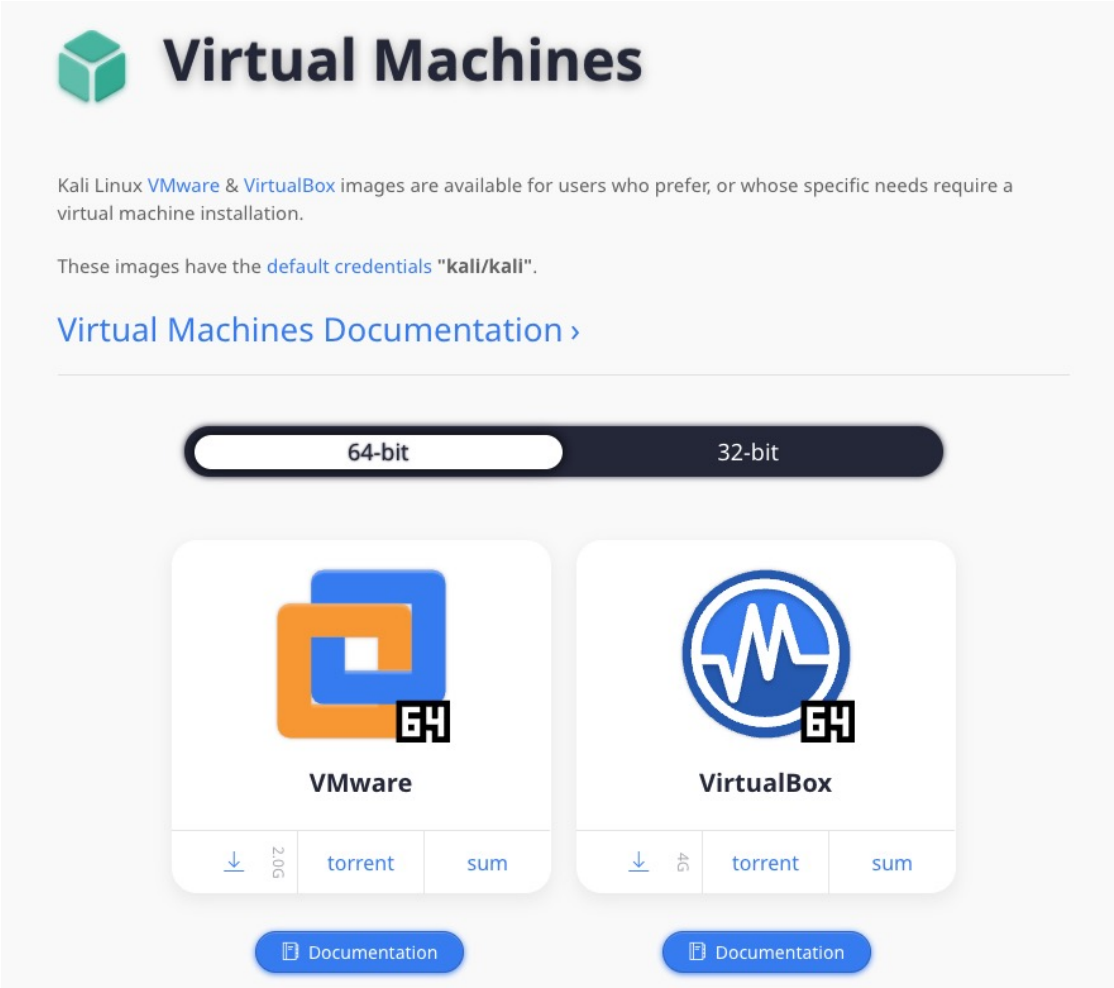
AGENDA

- Preparation for lab
- Get start with Kali Linux
- Introduction to Linux commands
- Shell script basic

PREPARATION FOR LAB

DOWNLOAD KALI

- The download URL for the Kali VM are as follows:
- URL: https://www.kali.org/get-kali/#kali-virtual-machines
- Recommend Kali Linux **VMware** image for this course
- You need 7-zip to unzip the file
- Kali VM default credentials:
  - Username: kali
  - Password: kali



DOWNLOAD VMWARE WORKSTATION

- VMware Workstation for Windows
  - VMware Workstation Player (Free)
  - https://www.vmware.com/hk/products/workstation-player/workstation-player-evaluation.html
  - VMware Workstation Pro (License)
  - https://www.vmware.com/hk/products/workstation-pro/workstation-pro-evaluation.html
- VMware Fusion for macOS (Intel CPU)
  - Register for Free 'Personal Use' License
  - https://www.vmware.com/hk/products/fusion/fusion-evaluation.html

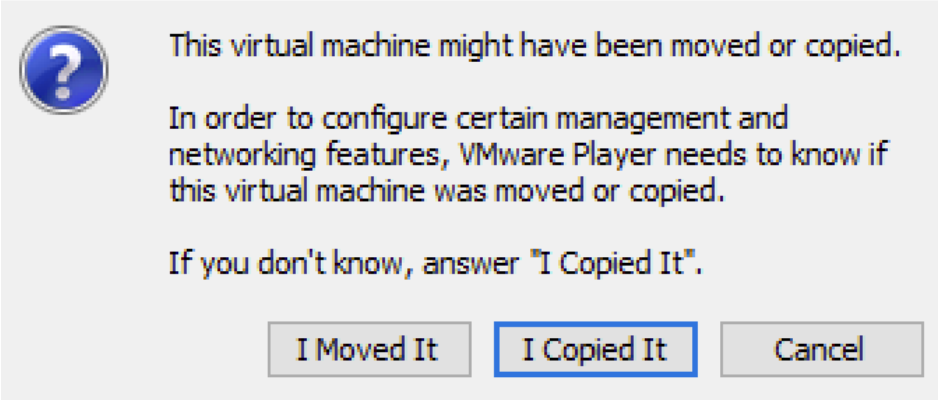
Fusion 12 Player for macOS 11+

REGISTER FOR A PERSONAL USE LICENSE ›

GET START WITH KALI LINUX

BOOT UP THE KALI VIRTUAL MACHINE

- Install VMWARE Workstation Pro/Player
- Unzip the Kali 7z archive with 7-Zip
- Navigate to the directory containing the extracted files
- Launch the VM by and double-click the .vmx file
- VMWARE will display a prompt asking if you moved or copied the VM. Click the "I Copied It" button to continue.
- Login the Kali VM
  - Username: kali
  - Password: kali



DOCKER INSTALLATION

$ sudo apt-get update

$ sudo apt-get install docker.io -y

$ sudo usermod -aG docker $USER

- Log out and log back in

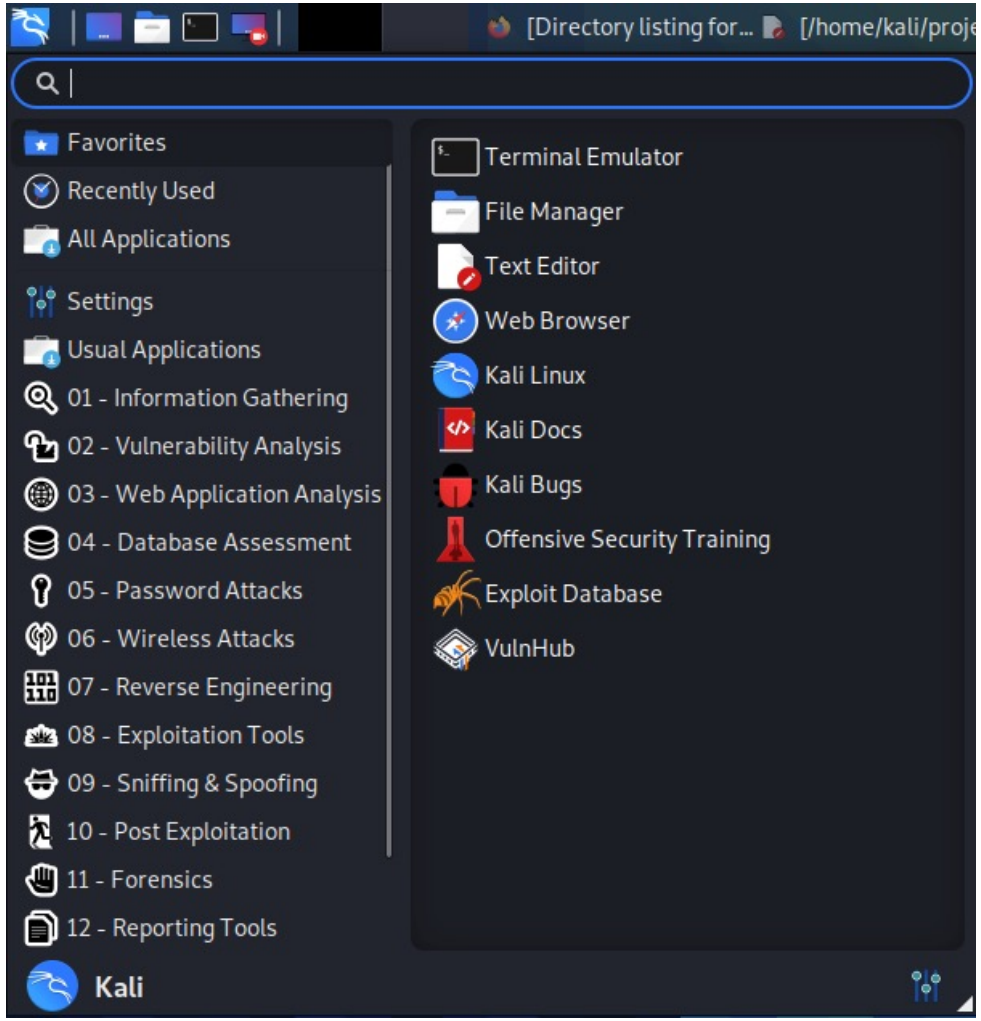
```
┌──(kali㉿kali)-[~]
└─$ docker version
Client:
 Version:           20.10.0+dfsg2
 API version:       1.41
 Go version:        go1.15.6
 Git commit:        7287ab3
 Built:             Mon Dec 14 12:39:22 2020
 OS/Arch:           linux/amd64
 Context:           default
 Experimental:      true

Server:
 Engine:
  Version:          20.10.0+dfsg2
  API version:      1.41 (minimum version 1.12)
  Go version:       go1.15.6
  Git commit:       eeddea2
  Built:            Mon Dec 14 12:39:22 2020
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.4.3~ds1
  GitCommit:        1.4.3~ds1-1+b1
 runc:
  Version:          1.0.0~rc92+dfsg1
  GitCommit:        1.0.0~rc92+dfsg1-5+b1
 docker-init:
  Version:          0.19.0
```

LAB TARGET VM – METASPLOITABLE 2

- Download target VM Metasploitable 2 Linux
  - https://sourceforge.net/projects/metasploitable/
- Launch the VM by and double-click the .vmx file
- Login the VM
  - Username: msfadmin
  - Password: msfadmin

LAB EXERCISE: EXPLORE THE KALI APPLICATIONS

- Spend your time to navigating the Applications Menu of Kali Linux
- There are tons of tools, they are categorized in the menu
- Take you time to familiar with the tools.
- You can find the user manual here https://www.kali.org/tools/



INTRODUCTION TO LINUX COMMANDS

LAB EXERCISE: EXPLORE LINUX COMMANDS

whoami	less
id	head
uname	ifconfig
passwd	grep
pwd	mkdir
cat	mv
cd	netstat
cp	rm
cut	find
chown	locate
chmod	which
echo	shutdown
ls	sort

To know more about the usage and options of a command, view its manual page

```
┌──(kali㉿kali)-[~]
└─$ man ip
```

```
IP(8)                                Linux                                IP(8)

NAME
       ip - show / manipulate routing, network devices, interfaces and tun-
       nels

SYNOPSIS
       ip [ OPTIONS ] OBJECT { COMMAND | help }

       ip [ -force ] -batch filename

       OBJECT := { link | address | addrlabel | route | rule | neigh |
               ntable | tunnel | tuntap | maddress | mroute | mrule | moni-
               tor | xfrm | netns | l2tp | tcp_metrics | token | macsec |
               vrf | mptcp | ioam }

       OPTIONS := { -V[ersion] | -h[uman-readable] | -s[tatistics] |
               -d[etails] | -r[esolve] | -iec | -f[amily] { inet | inet6 |
               link } | -4 | -6 | -B | -0 | -l[oops] { maximum-addr-flush-
               attempts } | -o[neline] | -rc[vbuf] [size] | -t[imestamp] |
               -ts[hort] | -n[etns] name | -N[umeric] | -a[ll] | -c[olor] |
               -br[ief] | -j[son] | -p[retty] }

OPTIONS
       -V, -Version
              Print the version of the ip utility and exit.
 Manual page ip(8) line 1 (press h for help or q to quit)
```

CHANGE THE DEFAULT ROOT PASSWORD

- The default password is *kali*. Use the passwd command to change the default password.
- Change the password before you start any services like SSH.

```
┌──(kali㉿kali)-[~]
└─$ passwd
Changing password for kali.
Current password:
New password:
Retype new password:
passwd: password updated successfully
```

FIND, LOCATE, AND WHICH

- These THREE commands can be used to locate files in the filesystem.
- locate - Before running locate you should update the local database using command “updatedb”
- which – Search through the directories that are defined in the $PATH
- find – A more aggressive search tool that able to recursively search any given path for various files

```
┌──(kali㉿kali)-[~]
└─$ locate plink.exe
/usr/share/windows-resources/binaries/plink.exe

┌──(kali㉿kali)-[~]
└─$ which plink.exe
plink.exe not found

┌──(kali㉿kali)-[~]
└─$ sudo find / -name plink*
/usr/share/windows-resources/binaries/plink.exe
find: '/run/user/1000/gvfs': Permission denied

┌──(kali㉿kali)-[~]
└─$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/local/games:/usr/games
```

MANAGING KALI SERVICES (SYSTEMCTL)

- Command “systemctl” start services
- Example:

  # systemctl start ssh

  # systemctl start apache2
- You can use “netstat” command to verify the service is running
- Example:

  # netstat –antp | grep sshd

  # netstat –antp | grep apache

```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl start ssh

┌──(kali㉿kali)-[~]
└─$ sudo netstat -antp | grep sshd
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1677/sshd: /usr
/sbi
tcp6       0      0 :::22                   :::*                    LISTEN      1677/sshd: /usr
/sbi
```

- You may want the service start automatically at boot time. Use “systemctl” to enable the services.
- Example:

  # systemctl enable ssh

  # systemctl enable apache2

SHELL SCRIPT BASIC (BASH/ZSH)

EXTRACT INFORMATION FROM FILES

- There is a HTML file “cisco-index.html” in the Lab1.zip.
- Your task is to find all subdomains in this HTML file.
- If you do it manually, it will be very time consuming. Using some bash commands can make the task easier.
- Looking over the file and find out the lines that contain the information we need and study the patterns.

```
<link rel="alternate" hreflang="zh-tw"
href="https://www.cisco.com/c/zh_tw/ind
ex.html"/>
```

```
        <li> <a href="https://learninglocator.cloudapps.cisco.com/
GlobalLearningLocator/LLocatorHome.do" data-config-metrics-group="
quick_tasks" data-config-metrics-title="prospects" data-config-met
rics-item="Learning" class="icon"><span class="center"></span></a>
```

GREP COMMAND

- We found that the lines contain URL must have the string “href=“.
- # **grep "href=" cisco-index.html**

```
┌──(kali㉿kali)-[~]
└─$ grep "href=" cisco-index.html
    <link rel="canonical" href="https://www.cisco.c
om"/>
        <link rel="alternate" hreflang="x-default"
href="https://www.cisco.com"/><link rel="alternate"
 hreflang="ja-jp" href="https://www.cisco.com/c/ja_
jp/index.html"/>
<link rel="alternate" hreflang="uk-ua" href="https:
//www.cisco.com/c/uk_ua/index.html"/>
```

CUT COMMAND

- We can see all the extracted line with similar structure.
- We can use cut command to split the line. If we use the "/" character as delimiter. The 3rd field is the subdomain.
- We use the pipe "|" to pass the grep command output as the input to cut command.

 # grep "href=" cisco-index.html |cut -d '/' -f 3

```
┌──(kali㉿kali)-[~]
└─$ grep "href=" cisco-index.html | cut -d '/' -f 3
www.cisco.com"
www.cisco.com"
www.cisco.com
```

CLEAN UP THE OUTPUT

- The output may contain some Non-URL entries.

```
375704031
etc
designs
www.cisco.com
```

- As we know the subdomain must contains a period "." character. We do "grep" command again.

 # grep "href=" cisco-index.html |cut -d '/' -f 3 |grep "\."

```
┌──(kali㉿kali)-[~]
└─$ grep "href=" cisco-index.html | cut -d '/' -f 3
|grep "\."
www.cisco.com"
www.cisco.com"
```

- Still the output is not perfect.

```
learningnetwork.cisco.com">Learning Network<
supportforums.cisco.com">Support Community<
```

- We can use the cut command again.

 # grep "href=" cisco-index.html | cut -d '/' -f 3 |grep "\." | cut -d '"' -f 1

```
┌──(kali㉿kali)-[~]
└─$ grep "href=" cisco-index.html | cut -d '/' -f 3
|grep "\." |cut -d '"' -f 1
www.cisco.com
www.cisco.com
www.cisco.com
```

SORT COMMAND

- The output contain a lot of duplicates.
- We can use sort command with the unique (-u) option

 # grep "href=" cisco-index.html |cut -d '/' -f 3 |grep "\." | cut -d '"' -f 1 | sort -u

```
┌──(kali㉿kali)-[~]
└─$ grep "href=" cisco-index.html | cut -d '/' -f 3
|grep "\." |cut -d '"' -f 1 |sort -u
blogs.cisco.com
blog.talosintelligence.com
com.cisco.androidcisco
communities.cisco.com
csr.cisco.com
developer.cisco.com
engage2demand.cisco.com
events-cisco.webex.com
```

REGULAR EXPRESSIONS IN GREP COMMAND

- The output still contains some entries are not valid subdomains

```
<link rel="alternate" href="android-app://com.cisco.androidcisco/c
isco.com/Home"/>
```

```
com.cisco.androidcisco
```

- We can use regular expressions in our command to extract the information.

 # grep -o 'https://[^"]*' cisco-index.html | cut -d "/" -f 3 | sort -u

```
┌──(kali㉿kali)-[~]
└─$ grep -o 'https://[^"]*' cisco-index.html|cut -d
 "/" -f 3 |sort -u
communities.cisco.com
engage2demand.cisco.com
events-cisco.webex.com
idreg.cloudapps.cisco.com
jobs.cisco.com
learninglocator.cloudapps.cisco.com
```

Regular expression reference: https://www.rexegg.com/regex-quickstart.html

REDIRECTING THE OUTPUT TO A TEXT FILE

- We can use “>” character in Shell (BASH / ZSH) to redirect the output.

```
┌──(kali㉿kali)-[~]
└─$ grep -o 'https://[^"]*' cisco-index.html|cut -d
 "/" -f 3 |sort -u > subdomains.txt

┌──(kali㉿kali)-[~]
└─$ wc -l subdomains.txt
18 subdomains.txt
```

FOR LOOP IN BASH

- Syntax
- # for var in 1 2 3; do echo $var; done
- # for var in $(cat file.txt); do echo $var; done

```
┌──(kali㉿kali)-[~]
└─$ for url in $(cat subdomains.txt ); do echo $url
; host $url |grep "has address" |cut -d " " -f 4; d
one
communities.cisco.com
13.226.226.19
13.226.226.100
13.226.226.24
13.226.226.82
engage2demand.cisco.com
142.0.160.17
events-cisco.webex.com
66.114.168.212
idreg.cloudapps.cisco.com
72.163.10.105
```

LAB EXERCISE: CREATE YOUR "DEMO.SH"

```
┌──(kali㉿kali)-[~/Desktop]
└─$ chmod +x demo.sh

┌──(kali㉿kali)-[~/Desktop]
└─$ ls -la
total 84
drwxr-xr-x  2 kali kali  4096 Jun 11 22:51 .
drwxr-xr-x 15 kali kali  4096 Jun 11 21:16 ..
-rw-r--r--  1 kali kali 71090 Jan 20  2021 cisco-index.html
-rwxr-xr-x  1 kali kali    69 Jun 11 22:50 demo.sh
```

```
┌──(kali㉿kali)-[~/Desktop]
└─$ ./demo.sh cisco-index.html
communities.cisco.com
engage2demand.cisco.com
events-cisco.webex.com
idreg.cloudapps.cisco.com
jobs.cisco.com
learninglocator.cloudapps.cisco.com
learningnetwork.cisco.com
locatr.cloudapps.cisco.com
marketplace.cisco.com
mycase.cloudapps.cisco.com
newsroom.cisco.com
search.cisco.com
secure.opinionlab.com
software.cisco.com
supportforums.cisco.com
twitter.com
www.cisco.com
www.schema.org
```

Learning shell scripting:
https://www.tutorialspoint.com/unix/shell_scripting.htm

COMP 7904
INFORMATION SECURITY: ATTACKS AND DEFENSE

LAB 1 PART 2 PASSWORD CRACKING

AGENDA

- cewl – Custom wordlist generator
- John the Ripper – Password Mutating
- Password Guessing and Password Cracking
- THC-Hydra
- John the Ripper – Password Cracking

CEWL - CUSTOM WORDLIST GENERATOR

- CeWL is a ruby app which spiders a given url to a specified depth, optionally following external links, and returns a list of words which can then be used for password crackers such as John the Ripper.
- cewl Usage Example
  - # cewl -d 2 -m 5 -w docswords.txt https://example.com
- Exercise: Create a wordlist with minimum word length of 9 against URL www.megacorpone.com and write the output to file “wordlist.txt”.

```
┌──(kali㉿kali)-[~]
└─$ sudo cewl -m 9 www.megacorpone.com -w wordlist.txt
[sudo] password for kali:
CeWL 5.4.8 (Inclusion) Robin Wood (robin@digi.ninja) (https://digi.ninja/)

┌──(kali㉿kali)-[~]
└─$ wc -l wordlist.txt
113 wordlist.txt
```

JOHN THE RIPPER - PASSWORD MUTATING

- We can use John The Ripper to mutate the password, include
  - Adding a few numbers at the end of the password
  - Swapping out lowercase of the capital letters
  - Changing certain letters to numbers
  - Etc.
- Study the mutating rules in /etc/john/john.conf file. You can add your own mutating rules inside.
- Example usage:

  # john --wordlist=wordlist.txt --rules --stdout > mutatedlist.txt

```
┌──(kali㉿kali)-[~]
└─$ john --wordlist=wordlist.txt --rule --stdout > mutatedlist.txt
Using default input encoding: UTF-8
Press 'q' or Ctrl-C to abort, almost any other key for status
5508p 0:00:00:00 100.00% (2021-01-23 23:10) 61200p/s Informationing

┌──(kali㉿kali)-[~]
└─$ wc -l mutatedlist.txt
5508 mutatedlist.txt
```

PASSWORD GUESSING VS PASSWORD CRACKING

- Password Guessing
  - Slow performance, depending to network and system response.
  - Generate large amount of network traffic and system logs.
  - Can cause Account Lockout / Deny of Service.
- Password Cracking
  - Better performance, no network and system restriction.
  - Required steal encrypted/hashed password from target system.
  - No Account Lockout / Deny of Services problem

THC-HYDRA

- Prepare an FTP Server for password guessing
- Start Metasploitable 2 Linux, VSFTP service is listening on TCP port 21
- Create FTP user

  $ sudo adduser comp7904
  - Password: football
- Check Metasploit 2's IP address

  $ ip addr
- Monitoring logs on Metasploitable 2

  # tail -f /var/log/auth.log

- Kali open xhydra
- Set the Single Target: {IP address of Metasploitable 2}
- Protocol: ftp
- Username: comp7904
- Password List: /usr/share/john/password.lst
- Monitoring network traffic via tcpdump
  - $ sudo tcpdump -nn -i eth0 dst port 21

JOHN THE RIPPER – CRACKING LINUX PASSWORD

- Linux system password file location
  - /etc/passwd
  - /etc/shadow
- The password algorithm varies depending on the different distribution and the version
  - MD5: 1
  - SHA-256: 5
  - SHA-512: 6

```
root@kali:~# cat /etc/shadow
root:$6$e0QZgb5q$73l4XlgU4XS.lv0UQ29R97M4H
```

SHA-512

salt

- Pseudo-random salt on Linux system
- unshadow – Combines passwd and shadow files

  # unshadow passwd shadow > unshadowed.txt
- john Usage Example

  # john --wordlist=/usr/share/john/password.lst --rules unshadowed.txt

  # john --show unshadowed.txt

COMP 7904
INFORMATION SECURITY: ATTACKS AND DEFENSE

LAB 1 PART 3 PASSIVE RECONNAISSANCE

AGENDA

- Introduction to Information Gathering
- OSINT
- Use Search Engines
- WHOIS & DNS Reconnaissance
- Use Social Networking Platforms
- Leverage OSINT Tools/Sites

INTRODUCTION

- It is important to know two types of reconnaissance (details ref to the lecture slides)
  - Passive Reconnaissance
    - OSINT - a subset of Reconnaissance
    - Our target doesn't know we are collecting their information
    - No/Low network traffic with the target
    - Keep ourselves stealthy -> Stealth is KEY ~ light touch / zero touch to your target!

  - Active Reconnaissance
    - Interact with the target system directly
    - Our footprint might leave on Network IDS (Intrusion and Detection System) and/or servers logs.

INTRODUCTION TO INFORMATION GATHERING

- Information gathering is the first and among the most critical step in your attack
- A strong phase of Information Gathering makes the difference between a good and a bad penetration tester
- Define an accurate scope, every information gathering stage will need the same focus and dedication as the first one
- Protect your own digital privacy
- ⚠️ Code of Ethics

WHAT IS OSINT

- Open-Source INTelligence (OSINT) is a method to collect and analyze data from open sources (overt and publicly available sources) to plan or take some action.
- Searching, Gathering, and Analyzing data found from public sources about your target.
- OSINT Source Examples
  - Search Engines (Google / DuckDuckGo / Baidu)
  - Dedicated Engines (Shodan / Have I Been Pwned)
  - Domain Registrars
  - Social Media (LinkedIn / Facebook / Instagram)
  - Pastes Sites (Pastebin)
  - Development Repository (GitHub)

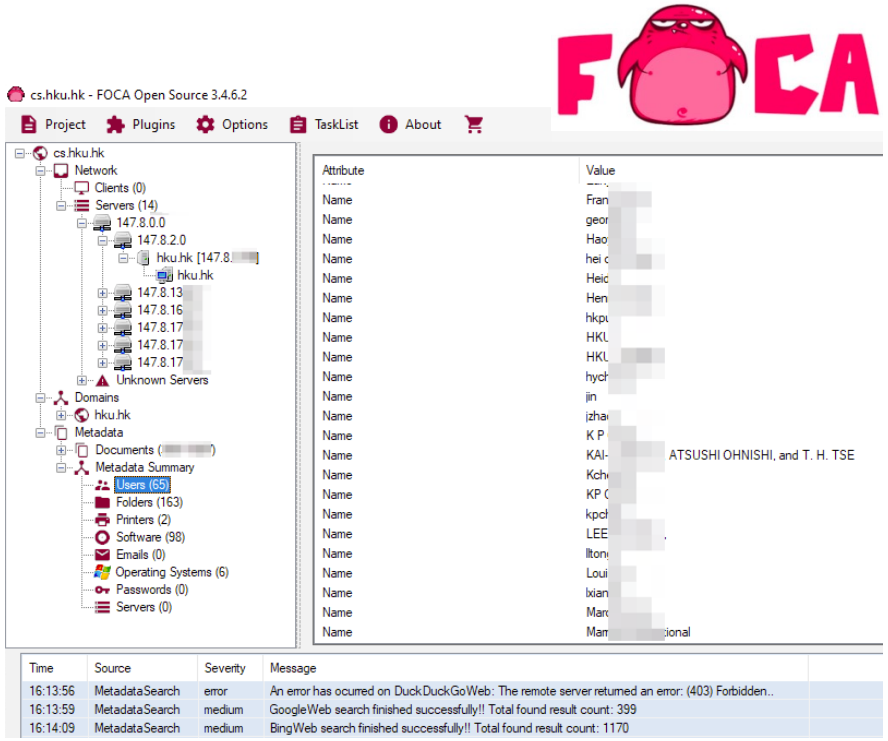
SEARCH ENGINES GOOGLE DORKING

- **Google Dorking**, also named **Google Hacking**, is a search technique that uses Google Search Engines to find security holes / perform the reconnaissance that websites are using.
- Basically, "Google hacking" involves using search operators in the Google search engine to locate specific text within search results, which the owner may doesn't aware of it.
- Example to finding specific versions of vulnerable Web applications
- Google Hacking Database (GHDB) - Google Dorks, OSINT, Recon https://www.exploit-db.com/google-hacking-database
- Beware google tracking your searches (Browser profiles, IP address, Geo)
- Ref: https://en.wikipedia.org/wiki/Google_hacking

- " " – put any phrase in quotes to use exact-match
- OR – defaults use logical AND between terms, specify "OR" (Capital Letter)
- - – excluded from the search result
- intitle: – search in page title
- inurl: – search in url
- intext: – search in body
- site: – search in domain
- cache: – search cached content
- filetype: / ext: – specific file type
- *Don't put spaces between the symbol or word and your search term.*

DOCUMENT METADATA FOCA

- Automates the process by **search document files and extracting their metadata**
- These documents are searched for using three possible search engines: Google, Bing, and DuckDuckGo.
- Support Windows Only: https://github.com/ElevenPaths/FOCA
- Required SQL / SQLExpress
- Integrates with Shodan (API Key is required) can helps to identify network ranges and additional targets



TOOLS
THEHARVESTER

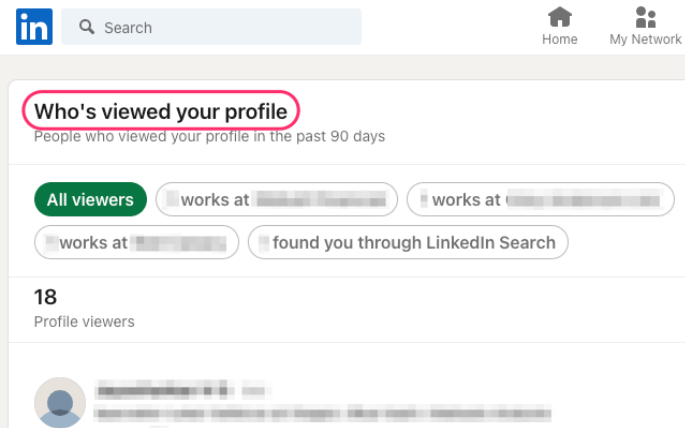
- The tool **gathers emails**, names, subdomains, IPs and URLs using multiple public data sources
- https://github.com/laramies/theHarvester
- # theHarvester.py -d hku.hk -l 100 -b google
  - -d: domain name
  - -l: limit the results (*** too many request will be blocked by search engine)
  - -b: the search engine (e.g. Google, Baidu, Bing, LinkedIn etc…)
  - -f: output to html file

ORGANIZATION RECON

- Business
- Project and Products
- Recent News
- Employee Name
- Position / Job Title
- Email Addresses
- Credentials

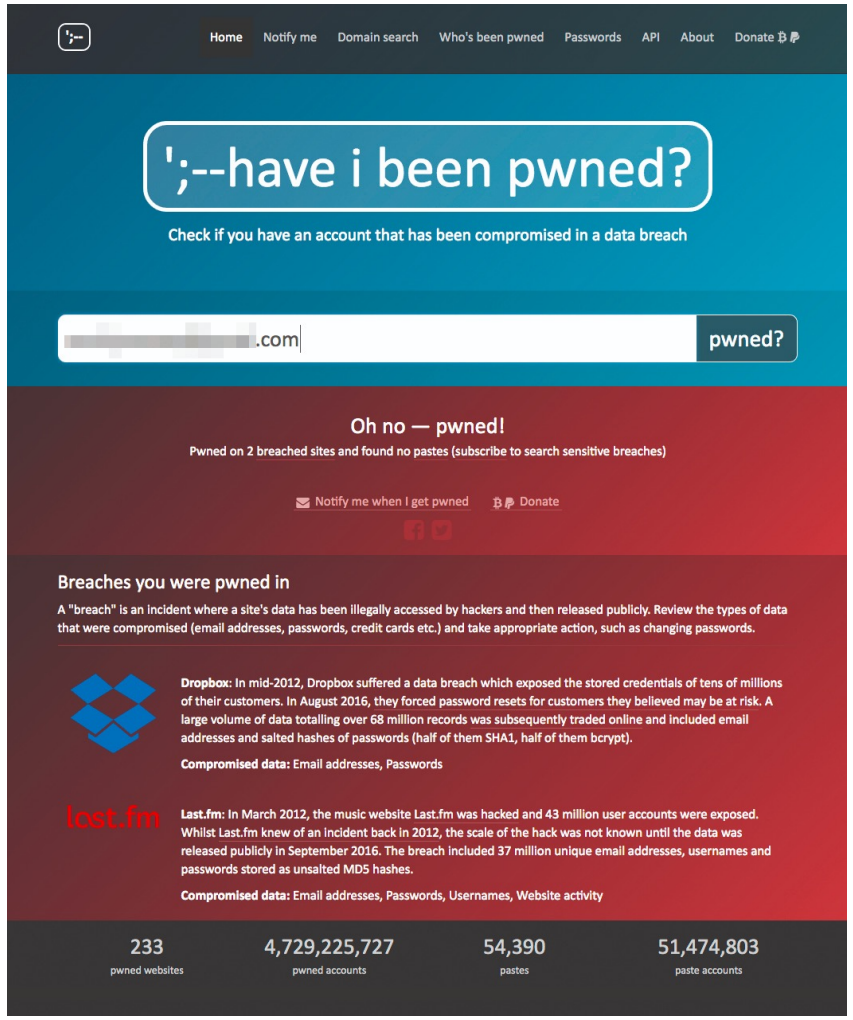
EMPLOYEES / ORGANIZATION INFORMATION

- LinkedIn is a great platform for performing reconnaissance against an organization.
- Often, you can build almost an entire organizational chart of every employee while gathering their names, job titles, and email addresses.
- Google's Mobile-Friendly Test
- LinkedInt is a tool for searching LinkedIn
  - Written by Vincent Yiu – an Offensive Security Expert in Hong Kong
  - https://github.com/vysecurity/LinkedInt
- Protect your own digital privacy
  - Burner account will work, but … burner account may have fewer results if it has small number of connections



PUBLIC DATA LEAKAGE

- **';--Have I Been Pwned?** (with "Pwned" pronounced like "poned,") is a website that allows internet users to check whether their personal data has been compromised by data breaches.
  - Have I Been Pwned? – https://haveibeenpwned.com
- Public data dump forums (many of them are seized by FBI in 2022)
- Torrents Leaked Database



PASTEBIN

- These items are examples of how paste sites are used by adverse hackers.
  - Email addresses and password lists
  - Login details
  - Stolen source code
  - Hacked data
  - Copyrighted information
  - Banking, credit card, or financial information
  - Personal information
  - Pornographic information
  - Spam links, including site promotion
- Pastebin doesn’t require user registration and allows for anonymous posting. This allows black hat hackers to easily and anonymously breach data in an accessible place.

INFRASTRUCTURE RECON

- Hostname, Domain Name, Subdomain Name
- IP addresses and subnet ranges
- Web Application and Technology in use
- Listening port and services
- Security control
- Services provider
- Vulnerabilities

WHOIS – DOMAIN NAME AND ITS OWNERSHIP

- Whois tool (whois client) is a query tool to query the whois server.
- Whois database contains **name server**, **registrar**, **contact information** about a domain name which is maintained by the domain name registrar. The central registry whois database is maintained by the InterNIC.
- Whois servers – Publish the whois databases, providing query service over TCP port 43.
- Web Based Whois Lookup: https://whois.domaintools.com/
- Lookup domain name

  # whois example.com
- Lookup IP address (reverse lookup)

  # whois 8.8.8.8

DNS

- DNS forward lookup
  - Domain name to IP address
- DNS reverse lookup
  - IP address to domain name
- DNS zone transfer
  - Get a list of domain names from a zone
- Public DNS Server
  - Google: 8.8.8.8, 8.8.4.4
  - Cloudflare: 1.1.1.1
  - Cisco Open DNS: 208.67.222.222, 208.67.220.220
  - Quad9: 9.9.9.9, 149.112.112.112

NS Record — Name Server

A Record — also known as a DNS host record, stores a hostname and its corresponding **IPv4** address.

AAAA Record — stores a hostname and its corresponding **IPv6** address.

CNAME Record — can be used to alias a hostname to another hostname.

MX Record — specifies an SMTP email server for the domain, used to route outgoing emails to an email server.

TXT Record — typically carries machine-readable data such as opportunistic encryption, sender policy framework, DKIM, DMARC, etc.

PTR Record — allows a DNS resolver to provide an IP address and receive a hostname (reverse DNS lookup).

SRV Record — a service location record, like MX but for other communication protocols.

DNS FORWARD LOOKUP

- Find the IP address of the host (A or AAAA record)
- You may need a list of common host names
  - www, mail, ns, mx, ftp…
- Command: host
  - # host www.example.com
- Command: nslookup
  - # nslookup www.example.com
- Command: dig
  - # dig www.example.com

```
root@kali:~# host www.example.com
www.example.com has address 93.184.216.34
www.example.com has IPv6 address 2606:2800:220:1:248:1893:25c8:1946
root@kali:~# host idontexist.example.com
Host idontexist.example.com not found: 3(NXDOMAIN)
root@kali:~# dig www.example.com

; <<>> DiG 9.11.4-2-Debian <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47963
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        5       IN      A       93.184.216.34
```

DNS REVERSE LOOKUP

- If the DNS administrator configured **PTR** records for the domain, we can perform the reverse lookup.
- And we might find out more domain names by probing the range of the found addresses.
- Command: host
  - # host 216.58.199.110

```
root@kali:~# host 216.58.199.110
110.199.58.216.in-addr.arpa domain name pointer hkg07s22-in-f14.1e100.net.
110.199.58.216.in-addr.arpa domain name pointer hkg07s22-in-f110.1e100.net.
```

DNS QUERY COMMAND

- Command: **host**
  - # host -t ns example.com
  - # host -t mx example.com
- Command: **dig**
  - # dig ns example.com
  - # dig mx example.com
  - # dig @8.8.8.8 example.com any +noall +answer +short
    - @8.8.8.8 use Google's DNS server
    - +noall turn off all contents
    - +answer turn on answer section
    - +short just show simple result

- Command: **nslookup** supported in Windows environment
- # nslookup
- Find A records
  - # set type=A
  - # example.com
- Find NS records
  - # set type=ns
  - # example.com
- Zone Transfer
  - # server [DNS server for example.com]
  - # ls –d example.com (Windows Only)

DNS ZONE TRANSFER (AXFR)

- DNS zone transfer should be limited to authorized slave DNS servers.
- But some DNS administrators misconfigure their DNS servers, anyone can get a copy of the DNS zone information by performing the zone transfer (AXFR).
- As a result, the hacker can get the network layout information form the misconfigured DNS server.
- Command: host
  - # host -l zonetransfer.me nsztm2.digi.ninja
- Command: dig
  - # dig axfr zonetransfer.me @nsztm2.digi.ninja

```
root@kali:~# host -t ns zonetransfer.me
zonetransfer.me name server nsztm2.digi.ninja.
zonetransfer.me name server nsztm1.digi.ninja.
root@kali:~# host -l zonetransfer.me nsztm2.digi.ninja
Using domain server:
Name: nsztm2.digi.ninja
Address: 34.225.33.2#53
Aliases:

zonetransfer.me has address 5.196.105.14
zonetransfer.me name server nsztm1.digi.ninja.
zonetransfer.me name server nsztm2.digi.ninja.
14.105.196.5.IN-ADDR.ARPA.zonetransfer.me domain name pointer www.zonetrans
fer.me.
asfdbbox.zonetransfer.me has address 127.0.0.1
```

DNS ENUMERATION

- Subdomain is shared the same top level domain name. e.g. cs.hku.hk, mail.hku.hk
  - Google Dorking – site:hku.hk -inurl:www
- DNSdumpster an open-source engine that can facilitate passive subdomain reconnaissance
  - DNSdumpster – https://dnsdumpster.com/
- AssetFinder
  - $ assetfinder -subs-only example.com
- SubFinder
  - https://github.com/projectdiscovery/subfinder
- Multi-threaded DNS recon tool
  - https://www.github.com/darkoperator/dnsrecon

```
root@kali:~# dnsrecon -d zonetransfer.me
[*] Performing General Enumeration of Domain: zonetransfer.me
[-] DNSSEC is not configured for zonetransfer.me
[*]      SOA nsztm1.digi.ninja 81.4.108.41
[*]      NS nsztm2.digi.ninja 34.225.33.2
[*]      Bind Version for 34.225.33.2 9.11.3-1ubuntu1.11-Ubuntu
[*]      NS nsztm1.digi.ninja 81.4.108.41
[*]      Bind Version for 81.4.108.41 9.10.3-P4-Debian
[*]      MX aspmx2.googlemail.com 108.177.8.26
[*]      MX alt2.aspmx.l.google.com 108.177.112.26
[*]      MX alt1.aspmx.l.google.com 108.177.8.26
[*]      MX aspmx.l.google.com 108.177.125.26
[*]      MX aspmx3.googlemail.com 108.177.112.27
[*]      MX aspmx5.googlemail.com 173.194.77.27
[*]      MX aspmx4.googlemail.com 172.253.112.27
[*]      MX aspmx2.googlemail.com 2607:f8b0:4003:c12::1b
[*]      MX alt2.aspmx.l.google.com 2607:f8b0:4001:c12::1a
[*]      MX alt1.aspmx.l.google.com 2607:f8b0:4003:c12::1a
[*]      MX aspmx.l.google.com 2404:6800:4008:c00::1a
[*]      MX aspmx3.googlemail.com 2607:f8b0:4001:c12::1b
[*]      MX aspmx5.googlemail.com 2607:f8b0:4023:401::1a
[*]      MX aspmx4.googlemail.com 2607:f8b0:4023::1b
[*]      A zonetransfer.me 5.196.105.14
```

SHODAN.IO

- Dedicated Engines – https://www.shodan.io/
  - Shodan performed the active scan for you, to grabs the resulting banners and scan the ports
  - **country:** two letter country code, e.g. HK
  - **city**: Search for results in a given city
  - **title**: Search the content scraped from the HTML tag
  - **html**: Search the full HTML content of the returned page
  - **product**: Search the name of the software banner
  - **version**: Search the version of the product
  - **hostname:** hostname or domain name, e.g. hku.hk
  - **net:** IP range or subnet
  - **os:** Operating Systems
  - **port:** not all ports are supported

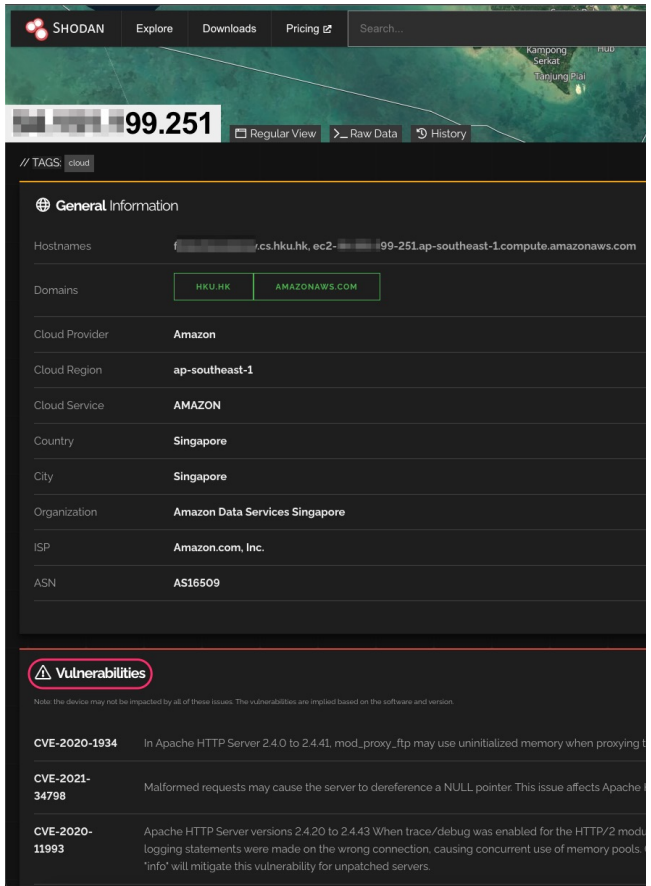
- Some filters are required a registered account

  Temporary Mailbox

  http://od.obagg.com/

  http://www.moakt.com/

Example: ***hostname:****cs.hku.hk* ***port:****443*



MALTEGO

- Maltego is an online intelligence gathering and visualization tools. Finding relationships between pieces of information from various online sources.
- Maltego automates the process of querying different data sources. This information is then displayed on a node-based graph suited for performing link analysis.
- Out-the-box Maltego comes with Machines for network footprinting, for example
  - **Footprint L1**: This is a basic footprint of a domain in its simplest form, lookup DNS server, IP address, etc.
  - **Company Stalker**: This option basically allows us to search email addresses, whois and social media networks.
- **Transforms** take pieces of information and use them to gather additional information.
- # apt install maltego

WEB DATA RECON

- Using Browser or Mobile Apps
- Virtual Browser
  - https://www.browserling.com
- Cached content from Search Engines / Cached Engines
  - Google Dorking – cache:hku.hk
  - Archive.org – http://archive.org
  - Archive.is – http://archive.is
  - Cached View – https://cachedview.com
- Determine a site is using a particular Web Technology
  - Urlscan.io – https://urlscan.io
  - Builtwith – https://builtwith.com
  - Netcraft – https://sitereport.netcraft.com
- CenSys.IO – https://search.censys.io
- WAF / CDN Detection
  - https://github.com/EnableSecurity/wafw00f

VULNERABILITIES INTELLIGENCE/DATABASE

- CVE Monitor – https://play8y3ar.github.io/cve_monitor
- CVE Trends – https://cvetrends.com
- Exploit DB – https://exploit-db.com
- Exploited in the wild – https://inthewild.io
- GitHub – https://github.com

SNIFFING TOOLS
TCPDUMP

- $ sudo tcpdump
  - -i [int]: Sniff on network interface
  - -n: Not resolve hostnames and services.
  - -v: Verbose
  - -X: Show contents in hexadecimal and ASCII.
  - -c: Number of packets to capture before stopping.
  - -e: Display Ethernet header data
  - -w: Save pcap file
  - -r: Read pcap file

FILTER

- Protocol
  - ether, ip, ipv6, arp, tcp, udp
- Type
  - host [hostname]
  - net [network]
  - port [portnumber]
  - portrange [start-end]
- Direction
  - src
  - dst
- Operator
  - and, or, not