388U Exam 1 Cheatsheet

Table of Contents

Lecture 1: Ethics

What is Ethical Hacking?

- Greyhat - good by day bad by night
- Whitehat - good
- Blackhat - bad
- "legally breaking into computer system with the express purpose of fixing the vulnerability

Legality VS Ethics

- They don't always overlap
- Something can be *legal* but **NOT** ethical and visa versa

Ethics on the Job

- Ethical hackers know...
  - The target, specifically the scope (IP addresses, URLS, etc)
  - Know the laws and the target's rules
  - Provide feedback
  - Minimize leftover exposure (clean up after yourself)
  - Act responsibly and carefully (demo/Proof of Concept instead of exploit)
- Non-disclosure agreements (NDAs)
  - Prevent pentesters or ethical hackers from talking about their work publicly if required by the client/target

Disclosure

- "Responsible disclosure": when you find a bug or a vulnerability you are ethically obligated to report it to the right party
- Where to report?
  - CISA (Cybersecurity and Infrastructure Security Agency)
  - Bug bounty platforms
    - Bugcrowd
    - HackerOne
  - Companies directly

Lecture 2: OSINT

What is OSINT?

Open Source Intelligence: basically using publicly available resources for

Active vs. Passive Recon

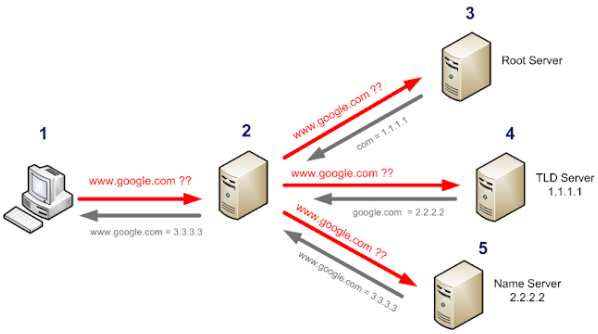
Active Recon: The attacker engages with the target system to gain more info *by interacting directly (You go get the data)*

Passive Recon: Attempt to gain info about target computers and networks without actively engaging with the systems (Data comes to you)

DNS

Domain Name Service:
Translates a domain name to an IP address

- DNS Dumpster: Gather information about domain names
- ICANN: Gather information about domain names
- `dig, whois, traceroute`



Web Source

Allows you to view the source code (HTML, CSS, and JS +others) of a webpage

- Good for finding links
- JavaScript code can contain vulnerabilities
- Find relevant libraries/services used by the website

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head profile="http://www.w3.org/2005/10/profile">
<!--<meta name="google-site-verification" content="ZbhFJOUU8lpryXEpJDWk2fXRThLcchuQpRuC4AMaRF4" />-->
<meta charset="utf-8"/>
<link rel="icon"
      type="image/png"
      href="https://registrar.umd.edu/images/master/favicon.ico" />
<link rel="stylesheet" type="text/css" href="https://www.registrar.umd.edu/css/xhtml-970px.css" />
<title>TESTUDO | Office of the Registrar</title>
<link rel="stylesheet" type="text/css" href="https://registrar.umd.edu/css/style.css" />
<script type="text/javascript">
function MM_swapImgRestore() { //v3.0
  var i,x,a=document.MM_sr; for(i=0;a&&i<a.length&&(x=a[i])&&x.oSrc;i++) x.src=x.oSrc;
}
function MM_preloadImages() { //v3.0
  var d=document; if(d.images){ if(!d.MM_p) d.MM_p=new Array();
    var i,j=d.MM_p.length,a=MM_preloadImages.arguments; for(i=0; i<a.length; i++)
    if (a[i].indexOf("#")!=0){ d.MM_p[j]=new Image; d.MM_p[j++].src=a[i];}}
}

function MM_findObj(n, d) { //v4.01
  var p,i,x;  if(!d) d=document; if((p=n.indexOf("?"))>0&&parent.frames.length) {
    d=parent.frames[n.substring(p+1)].document; n=n.substring(0,p);}
  if(!(x=d[n])&&d.all) x=d.all[n]; for (i=0;!x&&i<d.forms.length;i++) x=d.forms[i][n];
  for(i=0;!x&&d.layers&&i<d.layers.length;i++) x=MM_findObj(n,d.layers[i].document);
  if(!x && d.getElementById) x=d.getElementById(n); return x;
}

function MM_swapImage() { //v3.0
  var i,j=0,x,a=MM_swapImage.arguments; document.MM_sr=new Array; for(i=0;i<(a.length-2);i+=3)
   if ((x=MM_findObj(a[i]))!=null){document.MM_sr[j++]=x; if(!x.oSrc) x.oSrc=x.src; x.src=a[i+2];}
}
</script>

<style>
#top-menu{
    background: url("menu-bg.png") repeat scroll 0 0 transparent;
    left: -93px;
    height: 67px;
    position: absolute;
    top: 100px;
    width: 1160px;
}
#title {
    color: #fff;
    font-family: Serif72, Times, serif;
    font-size: 25px;
    text-align: left;
    overflow: hidden;
    text-overflow: ellipsis;
    -o-text-overflow: ellipsis;
}
#soclink {
    color:#cc0000;
    font-weight:bold;
}
```

robots.txt

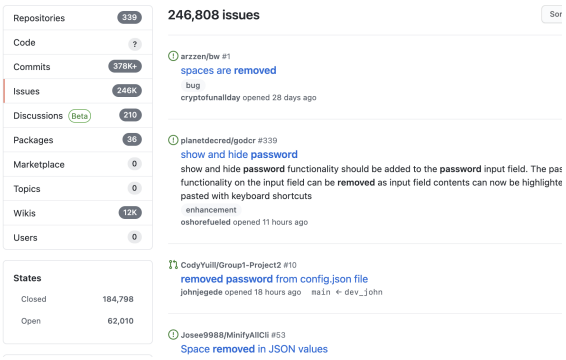
"Robots exclusion standard"

- Tells web crawlers where they can and can't look on a webpage
- Good place to start when trying to find important pages
- Web Crawlers (spiders)?
  - google.com, duckduckgo.com, etc

```
User-agent: *
Disallow: /search
Allow: /search/about
Allow: /search/static
Allow: /search/howsearchworks
Disallow: /sdch
Disallow: /groups
Disallow: /index.html?
Disallow: /?
Allow: /?hl=
Disallow: /?hl=*&
Allow: /?hl=*&gws_rd=ssl$
Disallow: /?hl=*&*&gws_rd=ssl
Allow: /?gws_rd=ssl$
Allow: /?pt1=true$
Disallow: /imgres
Disallow: /u/
Disallow: /preferences
Disallow: /setprefs
Disallow: /default
Disallow: /m?
Disallow: /m/
Allow:    /m/finance
Disallow: /wml?
Disallow: /wml/?
Disallow: /wml/search?
```

Git commits / Source

- Github is a site that allows programmers to post and collaborate on code
- People accidentally post passwords, API keys, etc
- Commit history is the previous changes that have been made
  - Will often contain secrets that weren't supposed to be seen

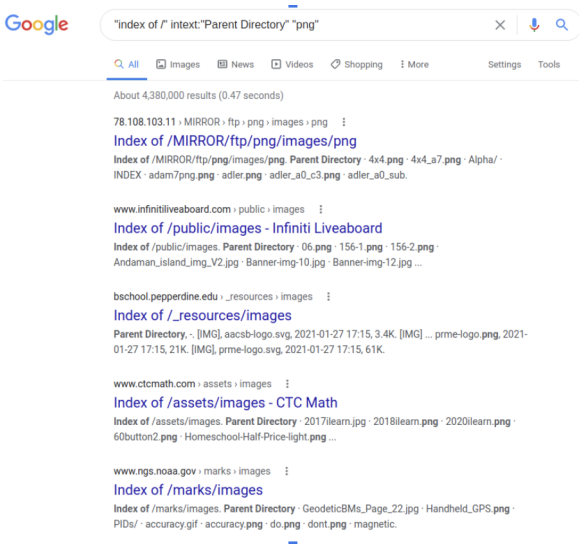


Google "dorking"

- Using google (or any search engines) advanced search

features to find interesting things

- https://www.exploit-db.com/google-hacking-database
- Lots of data!



Ports services

- Ports are a way to interface with a networked device (computer, IoT, etc)
- Services are the software that use a port to communicate with the internet
- Common ports and services
  - Port 22: SSH
  - Port 80: HTTP
  - Port 443: HTTPS
  - Port 25: SMTP

Shodan/Censys

- Companies that scan the internet or catalogue internet activity and provide that data
  - Show ports, grab banners, check services, sometimes scan for vulnerabilities
- https://shodan.io

- https://censys.io

Wayback Machine

- Huge internet archive of webpages, audio, video, text, etc.
- Can be used to see webpages in previous states
- Useful because you can watch changes in source code or services

Lecture 3: Vulnerability Scanning + OPSEC

Nmap

- "The Network Mapper"
  - Uber powerful tool that allows you to scan network hosts (both locally and on the internet)
  - Has Scripts!
  - Common CLI args/flags:
    - -p-, -A, -sV, -oN

SecList/Wordlists

- Categorized repository of security lists containing
  - Usernames / passwords
  - URLs
  - Fuzzing
  - Etc...
- Can be used by tools like `gobuster` or `dirbuster` to do directory bruteforcing, password bruteforcing, etc...

OPSEC

- "Operational Security"
- Security practices - both personal and at an organizational level
- Organizational
  - Controlled disclosure of information
  - Protects your org from data-breaches and potential exploitation or blackmail
- **Threat-modelling**: Finding balance between security and convenience; the more secure something is the odds are that the convenience is going to be limited (think VPN can result in slower internet connection)

Social Engineering

- Deceiving the target into providing you with information or taking an action
- **Pretexts:**
  - Art of creating an invented scenario to persuade a target into releasing information / performing an action
  - More than just lying...
    - Creating a thorough story that is believable enough to prevent exposure
    - Who are you pretending to be? Why are you calling/talking to the target? Why do you need the information you're asking for?

Lecture 4: Pentesting

Types of Pentests

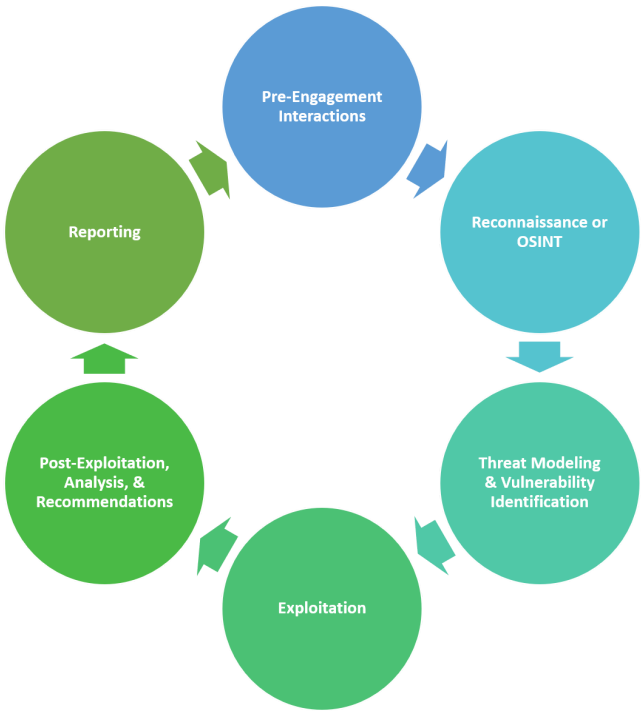
- **External Testing:** Testing assets that are visible on the internet

- **Internal Testing**: Testing assets *within* a target network (malicious insider)
- **Blind testing**: Tester is only given a limited amount of information (simulates real threat)
- **Double-Blind Testing**: Blue team (defense) doesn't know abt test + red team is blind
- **Targeted Testing**: Blue team and Red Team are working together at the same time

Pentest Categories

- **Physical**: Testing the physical security of a company / client
- **Network**: Breaking into and moving around a client network
- **Webapp**: Exploiting a webapp (facebook, gmail, etc) and using as foot hold to move farther
- **Hardware**: Breaking into physical devices (IoT)
- (*Non-exhaustive* but all you need to know for now)

Pentesting Methodology



CVEs

- **CVE (Common Vulnerabilities and Exposures):** A reference method for publicly known vulnerabilities (think dewey decimal system for hacks)
- **CVSS (Common Vulnerability Scoring System):** An open framework for communicating the characteristics and severity of software vulnerabilities
  - Based on a scale of 1-10 (where 10 is the scariest)
- https://cve.mitre.org/

Exploit Types

- **RCE (Remote Code Execution):** Command injection, buffer overflows, etc
- **Credential Disclosure:** Ability for unauthenticated user to see target credentials
- **DoS (Denial of Service):** Prevents proper use of a target usually by flooding the target

- **Authentication Bypass:** User can access system without the proper credentials
- **Info-leaks/Information Disclosure:** User can read information from a target they aren't supposed to have access too

Command Injection

- Unchecked/Unsanitized user input is passed to a `system() or exec()` there is the opportunity to allow for command injection
- `command1; command2` the semicolon separation allows for very basic command injection
- Remember that in bash if you aren't allowed to use spaces you can use the following syntax `{command1, argument1, argument2}` etc.

Privilege Escalation

- "Moving from a low value user to a high value one"
- Automatic tools:
  - LinPeas and WinPeas
  - LinEnum
- **SETUID Binaries**
  - Bit that you can set in file permission `chmod +u` which allows the binary to run with permissions different than the user
  - https://gtfobins.github.io/ (Linux)
  - https://lolbas-project.github.io/ (Windows)

Important Terms

- **Vulnerability:** is a weakness which can be exploited by a threat actor, such as an attacker, to cross privilege boundaries within a computer system or cause unexpected behavior

- **Exploit**: is a piece of software or a sequence of commands that takes advantage of a bug or vulnerability
- **0day vulnerability**: is a computer-software vulnerability that is unknown to those who should be interested in mitigating the vulnerability
- **backdoor**: refers to any method by which unauthorized users are able to get around normal security measures and gain high level user access

Lecture 5: Reverse Engineering

What is RE?

> The practice of analyzing a software system either in whole or in part to extract design and implementation information

What kinds of RE?

- **Hardware**: Figuring out how physical circuts/components work and interface
- **Firmware**:“Software the provides low-level control for a device’s specific hardware”
- **Software**: Looking at the inner working of software with express purpose of understanding of changing behavior

Software RE Basics

- Good first steps...
- `file` will give you information about a file type
- `strings` will output all **human readable** strings from a file

- `hexdump/bless` will give you raw bytes of a file

Static and Dynamic Analysis

- **Static Analysis**: Looking at an analyzing a programs source
  - Ghidra, jadx-gui, IDA, Binary Ninja, Hopper, etc
- **Dynamic Analysis**: Examining a program while its being run
  - Debuggers are primarily used for this
    - GDB and ProcDump
  - Radare2, angr, etc
- Summary: Dynamic analysis, you examine the behavior while running. Static analysis, you examine the code and try to figure out what it does

C Decompiling

- **Decompiling**: Takes the assembly code of a program, and attempts to reconstruct the original C code (or whatever other lang it was written in); can be inaccurate sometimes hence knowing the ASM will be helpful
- **Dissassembling**: Takes the raw machine code and translates it into assembly language to examine (pretty accurate)

Machine code → `disassembler` → **assembly** → `decompiler` → **c code**

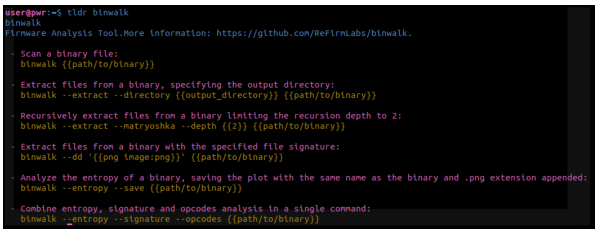
Java Disassembly

- Unlike C/C++ reversing, java can be disassembled more accurately
- **Smali**: Java equivalent of assembly, decompiled into source code

Firmware RE

Binwalk: is a firmware analysis tool (with many

applications outside of that). Allows you to locate and extract files using **magic bytes**.



binwalk help output

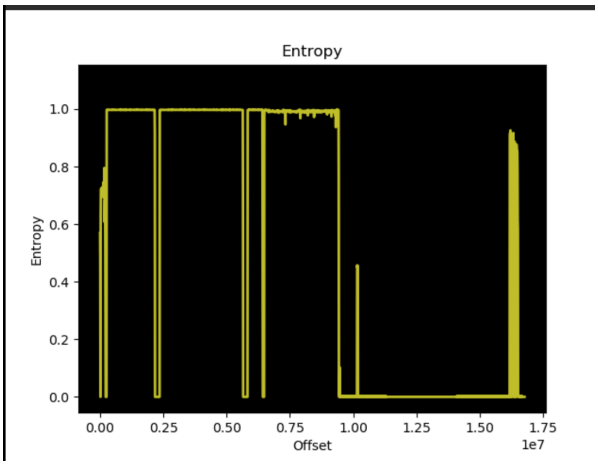
Magic bytes: are the indicators that computers use to read a file... (remember that extensions are arbitrary - the magic bytes are what define how a file can be interpreted... remember HW5 question 1.)

Executable Binaries	Mnemonic	Signature
DOS Executable	"MZ"	0x4D 0x5A
PE32 Executable	"MZ"...."PE.."	0x4D 0x5A ... 0x50 0x45 0x00 0x00
Mach-O Executable (32 bit)	"FEEDFACE"	0xFE 0xED 0xFA 0xCE
Mach-O Executable (64 bit)	"FEEDFACF"	0xFE 0xED 0xFA 0xCF
ELF Executable	".ELF"	0x7F 0x45 0x4C 0x46
Compressed Archives	**Mnemonic**	**Signature**
Zip Archive	"PK.."	0x50 0x4B 0x03 0x04
Rar Archive	"Rar!...."	0x52 0x61 0x72 0x21 0x1A 0x07 0x01 0x00
Ogg Container	"OggS"	0x4F 0x67 0x67 0x53
Matroska/EBML Container	N/A	0x45 0x1A 0xA3 0xDF
Image File Formats	**Mnemonic**	**Signature**
PNG Image	".PNG...."	0x89 0x50 0x4E 0x47 0x0D 0x0A 0x1A 0x0A
BMP Image	"BM"	0x42 0x4D
GIF Image	"GIF87a"	0x47 0x49 0x46 0x38 0x37 0x61
	"GIF89a"	0x47 0x49 0x46 0x38 0x39 0x61

sample magic bytes for files

File Entropy

- Basically - its the randomness within a file which can help one determine the state of the file (e.g is it encrypted? encoded? etc)
- How do you find file entropy?
  - `binwalk -E {filename}`



This is a sample file entropy graph

File Extraction

`binwalk -e {filename}` allows you to *carve* files using binwalk (or extract the secret files within a larger file like an IMG or a DMG)

Lecture 6: Forensics

What is digial forensics?

> Recovery, extraction, and analysis of data from a storage media

DFIR: Digital Forensics and Incident Response

- **Who and why?**
  - **Law Enforcement / Courts**: Used during investigations to recover information from suspect devices
  - **Malware analysts**: Figuring out **what** malware did on a system after infection
  - **Blue team/ SOC Operators**: Responding to security incidents
  - **Pentesters**: Extract information that could be used in an offensive capacity (Hashes, passwords, usernames, etc)

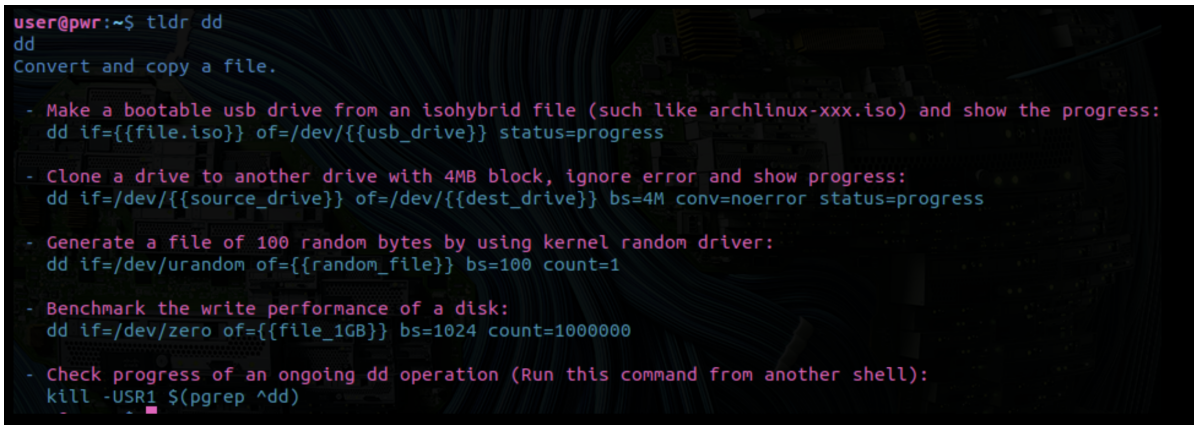
Data recovery from media

- Don't wanna just copy files! Trying to access information that has been deleted or damaged!
- Need to create an *image* or a *byte-for-byte* copy of the media
- Look but don't touch - collection now, analysis later
- **Common extensions**: img, iso, dmg, bin
- **Mediums to be analyzed**:
  - **Physical media**: HDDs, SSDs, thumbdrives, SIMS
  - **Storage Formats**: disk images, archives, logs

- **Live sources:** packet captures, memory dumps

Create Images

- `dd` is the main utility for creating images - you won't need to do this on the exam however its good to know generally



Copy and Verify

1. Create a verified copy of the image file
    - Create the copy using either `dd` or `cp`
    - Take a cryptographic hash of the original and compare to the copy (verified) to make sure that the copy is EXACTLY the same as the original
    - **never operate on the original file**

Hashing

- A one way cryptographic function that will easily demonstrate any changes made to a file... technicalities are not needed for the exam just know when they might be useful
- `md5sum {filename}`
- `sha256sum {filename}`
- `sha512sum {filename}`

Analysis Techniques

- Content-agnostic analysis
  - Using stuff from last lecture!
  - file, strings, hexdump, etc.
  - Generally less accurate, but more verbose
- Content-aware analysis
- Metadata analysis
- Steganographic analysis

Metadata

- "Information about the information"
- **Images (PNG/JPGS):**
  - Are especially interesting as they can contain...
    - GPS Tags
    - Time taken
    - Device information
    - Vendor tags
    - Software tags

Content-agnostic

- Same gist as RE tools - gaining an insight into the file using tools like `grep, strings, file, and binwalk`

Content-aware

- **Fingerprinting**: figure out who generated a file based on HOW they generated the file.
  - Compilers all have their own footprint, that footprint can be examined to attribute a file to a creator.
- **PDFs/Rich Documents**: embedded scripts, embedded change logs, default field values, references to other files.

- These file types give away lots of information about who/when/where the file was created using information stored within them (metadata on steroids)

Undelete

- Files are often not fully deleted... computers will often just dereference the information and wait for it to be overwritten... hence the information can be extracted before that point
- `extundelete` is the main tool packaged with kali linux used for this purpose
- *Important TLDR:* information stored on disk isn't fully deleted until it is overwritten with other information.

Steganography

> The practice of concealing information within other information

- NOT encryption, however can contain encrypted information

LSB Steganography

- 24-bit colorspace (8 bits per color) RGB
- 32-bit colorspace (8 bits per color) RGBA
- Hide information in the least significant bit of each color - since those changes are almost imperceptible to the human eye

Steghide

- Can be used to hide information in JPEG, BMP, WAV, AU files
- Files can be encrypted