✅Exam Cheat Sheet

Recovering Disk / MBR

🧪 Lab Link: INE

🔨 Tools:

- `HEX Workshop`
- `FTK Imager` / `Disk Editor`

📂Location: N/A - From an image file
#️⃣Syntax**:** GUI / Hex editor
🗒️Notes**:**

- The course taught us how to fix an MBR if the Signature is missing
- Correct the MBR by replacing tampered with bytes, prime example of this is the MBR Signature
  - `0x55aa`
  - If something else is corrupted in the MBR use Chat GPT but I don't think it will be, be careful not to go down a rabbit hole

Finding Hidden Partitions

A suspect may have made the partitions more difficult to find, but we can trace them down to find potentially hidden files
🧪 Lab Link: INE

🔨 Tool: `Autopsy / Disk Editor`

📂Location: N/A Based off disk image file
#️⃣**Syntax:** GUI - Instructions

Finding Hidden Partitions
Simply load it into Autopsy and look on the evidence tree for drives that are marked as `[Unallocated]` Especially if its a gap in between other partitions

Check Partitions are legit
A suspect may mess with the partitions and change signatures to be different **File Systems** or **Operating Systems** As such, check for key things within each drive like the root drive. E.g files that are like `$MFT` will only be in NTFS, not any other file systems

🗒️Notes:

Some useful observations are

Task	Key Observations	Tools & Methods
Identifying Disk Partitions	Check for **primary, extended, logical partitions**	**Autopsy, FTK Imager**
Detecting Partition Gaps	Gaps between partitions indicate **manual modification**	**Autopsy Hex editor**
Checking File System Mismatch	Partition type vs. actual file system **does not match**	**Autopsy**- Validate file system integrity
Correcting MBR Partition Entries	Modified partition types prevent OS detection	**Hex editor**- Restore correct partition values
Recovering Hidden Data	Data stored in **unallocated space or hidden partitions**	**Photorec, Foremost,** - File carving techniques
Analyzing Unallocated Space	Hidden or deleted files remain in slack space	**Bulk Extractor, Bstrings.exe** - Extract metadata and hidden strings
Detecting Malicious Disk Modifications	Suspicious edits to **MBR, GPT, partition table**	**Autopsy** - Detect unauthorized changes

Search For Hidden Files

We may need to find specific files that aren't immediately obvious, like based on file extension, file signature and more. Below should help make this a bit easier
🧪 Lab Link: N/A

🔨 Tool: `FTKImager`

📂Location: N/A
#️⃣**Syntax:** GUI - Instructions - See below for different ways to search

⚠️ Also try with other tools like Autopsy for deeper searches

Search Type	Method	Key Observations	Shortcut / Steps
File Browsing (Manual)	Expand evidence tree and navigate folders	Useful for structured file review	`File > Add Evidence Item` → Expand partitions
Search by File Name	Use the **Find** tool (`Ctrl + F`)	Finds exact or partial filenames	`File > Find` → Enter name → Click `Find Next`

Search by Extension	Sort by file type in the file list	Filters specific file formats like `.jpg`, `.pdf`	`View > File List` → Sort by `Extension`
Keyword Search (Text / Hex)	Searches within file contents	Identifies keywords inside text-based files	`File > Find` → Enter keyword → Select `Text` or `Hex`
Search by File Signature	Uses hash values to identify file types	Finds renamed or disguised files	`File > Export File Hash List` → Compare hashes
Search Deleted Files	Browse `$Recycle.Bin` or unallocated space	Recovers files even if not in directory listing	`Deleted Files` folder in FTK Imager
Search with File Filters	Uses pre-set filters like images, documents	Quickly narrows down to file types of interest	`View > Filter` → Select `Documents`, `Pictures`, etc.
Sort by File Metadata	Organize by date, size, attributes	Helps identify newly created, modified, or hidden files	`View > File List` → Sort columns (Date, Size, etc.)

Recover and Analyze Registry

🧪 Lab Link: INE

🔨 Tool: `Registry Explorer`

📂Location:
Syntax: `C:\Windows\System32\*`

- `NTUSER.Dat`
- `SAM`
- `SYSTEM`
- `SOFTWARE`
- `SECURITY`

#️⃣**Syntax:** GUI - Instructions

Load Offline Hive and open the **SYSTEM** file

> ⚠️ Be sure to check what is the active Registry Control Set, often its only one but could be trick question.
>
> Navigate through the Directory [File] > CsTool-CreateHive-etc > Select and look for the **Current** Value. This number will tell you what control set to look under.

🗒️**Notes:** Important things we can find are:

Category	What this is	Registry Location	Key
System Information	Computer Name	`SYSTEM\CurrentControlSet\Control\ComputerName\ActiveComputerName`	`ComputerName`
	Operating System	`SOFTWARE\Microsoft\Windows NT\CurrentVersion`	`ProductName / CurrentVersion`
	System Installation Date	`SOFTWARE\Microsoft\Windows NT\CurrentVersion`	`InstallDate`
	Registered Owner	`SOFTWARE\Microsoft\Windows NT\CurrentVersion`	`RegisteredOwner`

	System Root Directory	`SOFTWARE\Microsoft\Windows NT\CurrentVersion`	`SystemRoot`
	Last System Shutdown Time	`SYSTEM\ControlSet001\Control\Windows`	`ShutdownTime`
Time & Location Settings	Time Zone	`SYSTEM\CurrentControlSet\Control\TimeZoneInformation`	`TimeZoneKeyName / ActiveTimeBias`
	Daylight Saving Time (DST) Active?	`SYSTEM\CurrentControlSet\Control\TimeZoneInformation`	`DynamicDaylightTimeDisabled`
User & Authentication	Last Logged-in User	`SOFTWARE\Microsoft\Windows\CurrentVersion\Authentication\LogonUI`	`LastLoggedOnUser`
	Number of Users	`SAM\SAM\Domains\Account\Users`	`List of User SIDs`
	User SID and RID	`SAM\SAM\Domains\Account\Users\{User SID}`	`RelativeIdentifier (RID)`
	User Profile Creation Time	`SAM\SAM\Domains\Account\Users\{User SID}`	`CreatedTime`
	Last Logon Time	`SAM\SAM\Domains\Account\Users\{User SID}`	`LastLogon`
	Total Logon Count	`SAM\SAM\Domains\Account\Users\{User SID}`	`LogonCount`
	User Password Hint	`SAM\SAM\Domains\Account\Users\{User SID}`	`PasswordHint`
Network Information	Active Network Interface GUID	`SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Profiles`	`ProfileGuid`
	IP Address Assigned	`SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{GUID}`	`DhcpIPAddress`
	DHCP Name Server	`SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{GUID}`	`DhcpNameServer`
	Default Gateway	`SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{GUID}`	`DhcpDefaultGateway`
	DHCP Lease Obtained Time	`SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{GUID}`	`LeaseObtained`
	DHCP Lease Expiration Time	`SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{GUID}`	`LeaseExpires`
Security & Firewall	RDP Status (Firewall)	`SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicy\StandardProfile\GloballyOpenPorts`	???
Installed Applications	Installed Applications	`SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths`	`List of Installed Applications`
	Suspicious Remote Admin Tool (RAT)	`SOFTWARE\Microsoft\Windows\CurrentVersion\Run / RunOnce`	???

	Startup Applications	`SOFTWARE\Microsoft\Windows\CurrentVersion\Run / RunOnce`	`List of Startup Apps`
User Activity & Recent Files	Opened Documents (Recent)	`NTUSER.DAT\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSavePidlMRU`	`Recent File Paths`
	Last Opened File	`NTUSER.DAT\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs`	`Last Opened Document`
	Last Used Applications	`NTUSER.DAT\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\UserAssist`	`Most Recently Used Apps`
Mounted Devices	Mounted Devices	`SYSTEM\MountedDevices`	`List of Mounted Devices`
UserAssist & Usage Tracking	UserAssist Entries Count	`NTUSER.DAT\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\UserAssist`	`CEBFF5CD Subkey Count`
	UserAssist Encoding Type	`NTUSER.DAT\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\UserAssist`	`ROT13 Encryption`
	Most Executed Software	`NTUSER.DAT\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\UserAssist`	`Most Used Application`
Google Chrome Usage	Google Chrome Usage Count	`NTUSER.DAT\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\F4E57C4B-2036-3D9F`	`Chrome Shortcut Usage Count`
	Last Time Google Chrome Was Used	`NTUSER.DAT\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\F4E57C4B-2036-3D9F`	`Chrome Last Access Time`

Recover and Analyze Recycle Bin

🧪 Lab Link: INE

🔨 Tool: `FTKImager` and `rifiuti-vista.exe`

📂Location: `D:\$RECYCLE.BIN\S-1-5-21-[SID]\`

#️⃣**Syntax:** GUI - Instructions

Be sure to mount the Image file with FTKImager as a drive to be able to use `rifiuti-vista.exe` Then with **rifiuti** to analyse the bin

```
rifiuti-vista.exe D:\$RECYCLE.BIN\S-1-5-21-[SID]\
```

🗒️**Notes:** Be sure to check with FTKImager as well and compare results

Recover File Name based on $I and $R

Only Have the `$I` File

1. Locate the `$Ixxxxx` **file**.
2. Right-click the `$Ixxxxx` file and choose **View in Hex Viewer**.
3. The original filename and path are stored **starting at byte offset 0x18** (24 bytes in).
   - Scroll down in the hex view and look for **Unicode text** containing the **original file path and name**.

- The filename should be readable in **ASCII or UTF-16LE** format.

4. Throw into ChatGPT if its a LFN

If You Only Have the `$R` File

1. Identify the `$Rxxxxx` **file** which contains the **actual deleted file**.
2. Since the **original filename is not stored in the** `$R` **file itself**, you have two options:
   - **Find a matching** `$Ixxxxx` **file** (if available) in the same folder.
   - **Recover and open the file** to inspect its metadata.
     - Right-click the `$Rxxxxx` file and choose **Export**.
     - Use tools like **ExifTool (for images), PE headers (for executables), or file properties in Windows** to infer its original name.

Recover USB Disk traces

When someone has used a USB disk it leaves behind some traces of it being plugged in within the **registry**. We can find how many USBs have been connected and all their metadata

🧪 Lab Link: INE

🔨Tool: `Regripped`
📂Location / Files: `C:\Windows\System32\*`

- `NTUSER.Dat`
- `SAM`
- `SYSTEM`
- `SOFTWARE`
- `SECURITY`

Registry Locations

- `SYSTEM\CurrentControlSet\Enum\USBSTOR` - For Metadata of disks
- `NTUSER.DAT\Microsoft\Windows\CurrentVersion\Explorer\MountPoints` for what users accessed and mounted which disks

#️⃣Syntax: GUI

🗒️Notes:

- When ripped there are a few things to search for **MountedDevices**
- When looking for what users search the `ntuser.dat` file for **MountPoints2**

Analyze Windows Artifacts

⚠️ We may need to correlate multiple sources of evidence for one thing in particular, remember to check other sources if we were trying to determine something

🧪 Lab Link: INE

Searches

What searches and websites people have been doing

🔨**Tool:** `lecmd.exe`
📂**Location**: `%USERPROFILE%\AppData\Local\Microsoft\Windows\ConnectedSearch\History`

#️⃣**Syntax:** `.\LECmd.exe -d %USERPROFILE%\AppData\Local\Microsoft\Windows\ConnectedSearch\History`
🗒️**Notes:**

- `-d for directory -f for file`
- Look for Created / Modified Time and Search
- Title indicates Site title or search query
- _site files are websites visited
- _txt file are search queries

Opened Docs / lnk files 🔗

Recently opened files / folders

🔨Tool: `LECmd.exe`
📂Location: `%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\`
#️⃣Syntax: `.\LECmd.exe -d %USERPROFILE%\AppData\Local\Microsoft\Windows\Recent`

🗒️**Notes:**

- Same as above
- When a file is opened, it will create a lnk file in recent docs to the opened doc
- This can show where the lnk file points to

Shell bags (UsrClass.dat) 🔗

Keeps track of what folders people have opened, window and view preferences, window positions, sorting etc

🔨Tool: `shellbagsexplorer.exe`
📂Location: `%USERPROFILE%\AppData\Local\Microsoft\Windows\usrclass.dat`

#️⃣Syntax for CLI: `SBECmd.exe -d "%USERPROFILE%\AppData\Local\Microsoft\Windows\usrclass.dat" --csv Results\`

🗒️**Notes**

- Start with Shellbags explorer to get an idea of what was changed
- if needed correlate that with registry events that each shellbag will tell you its location
- Drag/ drop usrclass.dat into shellbags explorer
- Everything you see in Shellbags is what the user has accessed
  - look for Settings / Control panel, Downloads, and libraries to find sus stuff

Prefetch Files 🔗

Windows performance feature that helps programs start faster. When an application is launched, Windows creates a .pf (prefetch) file in: 📂 `C:\Windows\Prefetch\` We can track what programs are run and how many times

🔨Tool: `WinPrefetchView`
📂Location: `%windir%\prefetch`
#️⃣Syntax: GUI To make it analyze a different prefetch directory **Options** -> **Advanced Options** > **Path of Evidence**

🗒️Notes:

- By tracking what files are loading at start up this helps us find sus programs, malware and external devices
- Hash value is in the exe name e.g zenmap.exe-[**hashvalue**].pf

Thumbnails / Cache 🔗

This tracks previews of images that have been opened by a user, so if theyve deleted it, evidence may still remain

🔨Tool: `thumbcache_viewer`
📂Location: `%USERPROFILE%\AppData\Local\Microsoft\Windows\Explorer`
#️⃣Syntax: GUI

🗒️Notes:

- Can open multiple db files all at once
- Manually process, just browse through

Jump list

Jump Lists are a Windows feature that tracks recently opened files, folders, and websites for specific applications (Word Excel etc). They allow users to quickly reopen recent items from the taskbar or Start menu. Files here have a **.automaticDestinations-ms** or **.customDestinations-ms** extension.

🔨Tool: `JumpListExplore`
📂Location: `%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\CustomDestinations` `%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations`
#️⃣Syntax: GUI

🗒️Notes: Double click on an item to get its full path, combine Local Path and Common path to make the full directory

Pinned Out will show if its pinned to the taskbar

Created dates and Last modified dates here are important as it helps with timelines as well as shows if something has been opened more than once

User Libraries

User Libraries in Windows are virtual collections of folders that help organize and access files efficiently. Unlike normal folders, Libraries don't store files directly but instead link to multiple locations.

🔨Tool: `Notepad / Hex editor`
📂Location: `%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Libraries`
#️⃣Syntax: GUI

🗒️Notes: Library files are XML files that keep track of what folders to include in the library. We can track sus user activity from looking at Library files, they might reveal directories that have since been deleted.

Analyze Network Attacks

try to find what network based attacks and compromised hosts by looking at PCAP Files.

🧪 Lab Line: INE
🔨 Tool: `Wireshark`
#️⃣ Syntax: GUI / Query based examples below

🗒️Notes: List of queries and attacks to search for follows

Network Attacks

Attack Type	Detection Method	Key Indicators	Wireshark Filters
MAC Flooding	Check for excessive random MAC addresses	High volume of packets from unknown MACs	`eth.addr` with high variation
ARP Poisoning	Inspect ARP replies without corresponding requests	Conflicting MAC addresses for gateway	`arp` filter, `arp.opcode == 2` (unsolicited replies)

SYN Flood (DoS)	Look for excessive SYN requests with no ACK	Unfinished TCP handshakes, high SYN packet rate	`tcp.flags.syn == 1 && tcp.flags.ack == 0`
DNS Amplification (DDoS)	Check for large DNS responses to small requests	High volume of DNS responses with large payloads	`dns` filter, `udp.length > 512`
ICMP Flood	Monitor excessive ICMP Echo requests	Continuous pings without replies	`icmp.type == 8` (Echo Request)
DHCP Starvation	Look for excessive DHCP Discover packets	Clients stuck with APIPA addresses, no assigned IPs	`dhcp.option.dhcp_message_type == 1`
Rogue DHCP Server	Identify multiple DHCP Offer messages	Conflicting DHCP servers providing IPs	`dhcp.option.dhcp_message_type == 2`
Man-in-the-Middle (MITM)	Check ARP, DNS, or SSL traffic for anomalies	Unexpected ARP replies, certificate mismatches	`arp` filter, `ssl.alert_message`
Port Scanning	Look for sequential connection attempts on ports	Multiple SYN packets to different ports	`tcp.flags.syn == 1 && tcp.flags.ack == 0`
SMB Relay Attack	Check for NTLM authentication attempts over SMB	SMB authentication requests from unexpected sources	`smb` filter, `ntlmssp.auth`
SNMP Enumeration	Look for excessive SNMP Get requests	Unauthorized SNMP queries on UDP 161	`udp.port == 161`
Rogue Access Point (Wi-Fi)	Identify duplicate SSIDs or unauthorized BSSIDs	Unexpected APs with same SSID but different MACs	`wlan.ssid` with multiple BSSIDs
Packet Injection	Analyze unexpected packet types	Malformed or out-of-sequence packets	`tcp.analysis.flags`
Remote Access Backdoor	Look for unusual outbound traffic patterns	Connections to unknown IPs on non-standard ports	`tcp.port == 6666`, `icmp contains "data"`

Web Attacks

Attack Type	Detection Method	Key Indicators	Wireshark Filters
SQL Injection (SQLi)	Analyze HTTP logs for SQL commands	`' OR 1=1 --`, `UNION SELECT`, database errors	`http.request.uri contains "SELECT"`
Local File Inclusion (LFI)	Look for file path traversal attempts	`../../etc/passwd`, `%2e%2e%2f`	`http.request.uri contains "../"`
Remote File Inclusion (RFI)	Check for external script execution	`http://malicious.com/shell.txt`	`http.request.uri contains "http://"`

Directory Traversal	Look for encoded directory traversal patterns	`/etc/passwd`, `../win.ini`, `%2e%2e%2f`	`http.request.uri contains "../"`
Cross-Site Scripting (XSS)	Check for `<script>` tags and encoded payloads	`<script>alert('XSS')`, `%3Cscript%3Ealert(1)`	`http.request.uri contains "<script>"`
Command Injection	Look for shell command separators	`; cat /etc/passwd`, `&& whoami`, `cmd.exe /c dir`	`http.request.uri contains "&&"`
HTTP Header Attacks	Inspect `Host`, `X-Forwarded-For`, `User-Agent`	`Host: attacker.com`, `curl/7.68.0`, spoofed IPs	`http.header contains "X-Forwarded-For"`
SYN Flood (DoS Attack)	High volume of SYN packets with no ACK	SYN requests only, no handshake completion	`tcp.flags.syn == 1 && tcp.flags.ack == 0`
DHCP Starvation	Excessive DHCP Discover packets	Clients stuck with APIPA addresses	`dhcp` filter, `dhcp.option.dhcp_message_type == 1`
Brute Force / Credential Stuffing	Multiple failed logins	Excessive `401 Unauthorized`, repeated login attempts	`http.request.method == "POST" and http.request.uri contains "login"`
Web Shell Detection	Suspicious PHP uploads & unusual response content	`cmd.php`, `shell.php`, unexpected `uid=0(root)`	`http.request.uri contains ".php"`
Remote Access Backdoors	Unknown protocol or port usage	`tcp.port == 6666`, connections over ICMP	`tcp.port == 6666`, `frame contains "data"`