

[Open in app ↗](#)**Medium**

Search



★ Get unlimited access to the best of Medium for less than \$1/week. [Become a member](#)



# TryHackMe Critical Write-Up: Using Volatility For Windows Memory Forensics



Joseph Alan · Follow

Published in System Weakness

5 min read · Jul 18, 2024

[Listen](#)[Share](#)[More](#)

This challenge focuses on memory forensics, which involves understanding its concepts, accessing and setting up the environment using tools like Volatility, gathering information from the compromised target, searching for suspicious activity with the obtained data, and extracting and analyzing information from memory dumps using various Volatility plugins.

## Task 1 Introduction

### Learning Objectives

In this room, we'll cover the following learning objectives.

- Memory Forensics' basic concepts.
- How to access and set up the environment.
- Gathering information from the compromised target.
- Search for suspicious activity using the information obtained.
- Extracting and analyzing data from memory.
- Conclusion & further steps after completing the room.

## Task 2 Memory Forensics

- Memory forensics is a subset of computer forensics that analyzes volatile memory, typically on a compromised machine.
- In Windows OS, this corresponds to the Random Access Memory (RAM), whose content is flushed with every reboot or shutdown.
- It's one of the usual initial tasks during an incident response.
- **Difference from Disk Forensics:**
  - Provides information about what resides on the target computer.
  - Offers details on running processes or applications at a specific time.
  - Gives insights into execution flow that may not be present in regular storage units or application logs.
- **Immediate Snapshot:** Memory analysis can provide an immediate snapshot of an application or a timestamp of an attacker's actions.

- **Chronology of Events:** Evidence collected through memory forensics is crucial for creating a timeline of events.
- **Two Main Phases:**

## 1. Memory Acquisition:

- Copying live memory to a file (dump).
- Essential to perform analysis without risking data loss from an inadvertent reboot.
- Provides proof of analysis if needed.

## 2. Memory Analysis:

- Detailed examination of the acquired memory dump to extract useful information.

## Task 2 Answers

Answer the questions below

---

What type of memory is analyzed during a forensic memory task?

RAM

✓ Correct Answer

In which phase will you create a memory dump of the target system?

Memory Acquisition

✓ Correct Answer

## Task 3 Environment & Setup

### Imaging Tools

There are several ways to acquire the memory from the target machine if needed; several tools can help us, but which one to use will depend on personal preference and the OS involved in the imaging task. Some of these tools are:

Windows	<a href="#">FTK imager, WinPmem</a>
Linux	<a href="#">LIME</a>
macOS	<a href="#">osxpmem</a>

In our scenario, [FTK Imager](#) was used to take the memory dump of the compromised machine, which was copied to the [Linux](#) machine to perform the analysis.

## Accessing the Machine

Before moving forward, start the lab by clicking the Start Machine button. It will take around 2 minutes to load properly. The VM will be accessible on the right side of the split screen. In case the VM is not visible, use the blue Show Split View button at the top of the page. You can also connect directly to the machine using the following information via SSH:



A memory dump named `mendump.mem` will be present at the home address at `/home/analyst`

Volatility3

```
user@machine$ vol -h
usage: volatility [-h] [-c CONFIG] [--parallelism [{processes,threads,off}]] [-v]
                  [-l LOG] [-o OUTPUT_DIR] [-q] [-r RENDERER] [-f FILE]
                  [--clear-cache] [--cache-path CACHE_PATH] [--offline] [--sing
                  [--stackers [STACKERS [STACKERS ...]]]
                  [--single-swap-locations [SINGLE_SWAP_LOCATIONS [SINGLE_SWAP_
```

## For Windows

```
user@machine$ vol windows --help
usage: volatility [-h] [-c CONFIG] [--parallelism [{processes,threads,off}]] [-e EXTEND] [-p PLUGIN_DIRS] [-s SYMBOL_DIRS] [-v] [-l LOG]
                  [-o OUTPUT_DIR] [-q] [-r RENDERER] [-f FILE] [--write-config] [--save-config SAVE_CONFIG] [--clear-cache] [--cache-path CACHE_PATH] [--offline]
                  [--single-location SINGLE_LOCATION] [--stackers [STACKERS [STACKERS ...]]] [--single-swap-locations
                  [SINGLE_SWAP_LOCATIONS [SINGLE_SWAP_LOCATIONS ...]]]
                  plugin ...
volatility: error: argument plugin: plugin windows matches multiple plugins (windows.bigpools.BigPools, windows.cachedump.Cachedump,
windows.callbacks.Callbacks, windows.cmdline.CmdLine, windows.crashinfo.Crashinfo, windows.devicetree.DeviceTree,
windows.dlllist.DllList, windows.driverIrp.DriverIrp, windows.drivermodule.DriverModule, windows.driverscan.DriverScan,
windows.dumpfiles.DumpFiles, windows.environ.Envvars, windows.filescan.FileScan, windows.getservicesids.GetServiceIDs,
windows.getsids.GetSIDs, windows.handles.Handles, windows.hashdump.Hashdump, windows.info.Info, windows.joblinks.JobLinks,
windows.ldrmodules.LdrModules, windows.lsadump.Lsadump, windows.malfind.Malfind, windows.mbrscan.MBRScan, windows.memmap.Memmap,
windows.mftscan.ADS, windows.mftscan.MFTScan, windows.modscan.ModScan, windows.modules.Modules, windows.mutantscan.MutantScan,
windows.netscan.NetScan, windows.netstat.NetStat, windows.poolscanner.PoolScanner, windows.privileges.Privs, windows.pslist.PsList,
windows.psscan.PsScan, windows.pstree.PsTree, windows.registry.certificates.Certificates, windows.registry.hivelist.HiveList,
windows.registry.hivescan.HiveScan, windows.registry.printkey.PrintKey, windows.registry.userassist.UserAssist,
windows.sessions.Sessions, windows.skeleton_key_check.Skeleton_Key_Check, windows.ssdt.SSDT, windows.statistics.Statistics,
windows.strings.Strings, windows.svcskan.SvcScan, windows.symlinkscan.SymlinkScan, windows.vadinfo.VadInfo, windows.vadwalk.VadWalk,
windows.vadyarascan.VadYaraScan, windows.verinfo.VerInfo, windows.virtmap.VirtMap)
```

Plugins are extremely helpful during the analysis when using Volatility3 since they will quickly parse a memory dump for specific data types and sort the data according to the selected plugin. You can find a summary of some of the most relevant plugins below

Windows.cmdline	Lists process command line arguments
windows.drivermodule	Determines if any loaded drivers were hidden by a rootkit
Windows.filescan	Scans for file objects present in a particular Windows memory image
Windows.getsids	Print the SIDs owning each process
Windows.handles	Lists process open handles
Windows.info	Show OS & kernel details of the memory sample being analyzed
Windows.netscan	Scans for network objects present in a particular Windows memory image
Windows.netstat	Traverses network tracking structures present in a particular Windows memory image.
Windows.mftscan	Scans for Alternate Data Stream
Windows.pslist	Lists the processes present in a particular Windows memory image
Windows.pstree	List processes in a tree based on their parent process ID

## Task 3 Answers

Answer the questions below

Which plugin can help us to get information about the OS running on the target machine?

Windows.info

✓ Correct Answer

Which tool referenced above can help us take a memory dump on a Linux OS?

LIME

✓ Correct Answer

Which command will display the help menu using Volatility on the target machine?

vol -h

✓ Correct Answer

## Task 4 Gathering Target Information

Using `-f` switch to indicate the file to analyze, in this case, `memdump.mem` followed by the plugin `windows.info` used to get the general information, as in the example shown below.

```

1 2 | analyst@ip-10-10-19-67: ~
root@kali:/home/kali/Downloads
analyst@ip-10-10-19-67:~$ vol -f memdump.mem windows.info
Volatility 3 Framework 2.5.2
Progress: 100.00          PDB scanning finished
Variable           Value
Kernel Base        0xf8066161b000
DTB               0x1ad000
Symbols file:///home/analyst/volatility3-2.5.2/volatility3/symbols/windows/ntkrnlmp.pdb/4DBE144182FF4156845CD3BD8B654E56-1.json.xz
Is64Bit True
IsPAE  False
layer_name        0 WindowsIntel32e
memory_layer      1 FileLayer
KdVersionBlock   0xf8066222a400
Major/Minor       15.19041
MachineType      34404
KeNumberProcessors 2
SystemTime        2024-02-24 22:52:52
NtSystemRoot      C:\Windows
NtProductType    NtProductWinNT
NtMajorVersion   10
NtMinorVersion   0
PE MajorOperatingSystemVersion 10
PE MinorOperatingSystemVersion 0
PE Machine        34404
PE TimeStamp      Sat Jan 13 03:45:32 2085
analyst@ip-10-10-19-67:~ |

```

The windows.info plugin provides info about the OS

Kernel Base Address

64-bit arch is true

OS version

## Task 4 Answers

Answer the questions below

Is the architecture of the machine x64 (64bit) Y/N?

Y

✓ Correct Answer

✗ Hint

What is the Version of the Windows OS

10

✓ Correct Answer

✗ Hint

What is the base address of the kernel?

0xf8066161b000

✓ Correct Answer

✗ Hint

## Task 5 Searching For Suspicious Activity

Using the windows.netscan plugin

```
analyst@ip-10-10-19-07:~$ volatility -f memdump.mem windows.netscan --prep 80
0x50ed0d0de60_0TCPv4 192.168.182.139:49747fin13.107.42.254 443 CLOSED 4780 SearchApp.exe 2024-02-24 22:50:42.000000
0x50ed0d7ef8100_0DPv4 192.168.182.139 137 * 0 4 System svchost.exe 2024-02-24 22:47:36.000000
0x50ed0d54060_0DPv4 0.0.0.0 0 * 0 1360 svchost.exe 2024-02-24 22:47:36.000000
0x50ed0d54060_0DPv6 :: 0 * 0 1360 svchost.exe 2024-02-24 22:47:36.000000
0x50ed0d54510_0DPv4 0.0.0.0 5353 * 0 1360 svchost.exe 2024-02-24 22:47:36.000000
0x50ed0d54510_0DPv6 :: 5353 * 0 1360 svchost.exe 2024-02-24 22:47:36.000000
0x50ed0d56c20_0DPv4 0.0.0.0 5353 * 0 1360 svchost.exe 2024-02-24 22:47:36.000000
0x50ed0d83090_0DPv4 0.0.0.0 5355 * 0 1360 svchost.exe 2024-02-24 22:47:36.000000
0x50ed0d83090_0DPv6 :: 5355 * 0 1360 svchost.exe 2024-02-24 22:47:36.000000
0x50ed0d83090_0DPv6 0.0.0.0 5355 * 0 1360 svchost.exe 2024-02-24 22:47:36.000000
0x50ed0d83090_0TCPv4 0.0.0.0 7680 0.0.0.0 0 LISTENING 5572 svchost.exe 2024-02-24 22:47:36.000000
0x50ed0d818e0d0_0TCPv4 192.168.182.139 49763 23.222.237.203 443 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:50:39.000000
0x50ed0d818e0d0_0TCPv6 :: 7680 :: 0 LISTENING 5572 svchost.exe 2024-02-24 22:47:36.000000
0x50ed0d83e4d0_0TCPv4 0.0.0.0 5840 0.0.0.0 0 LISTENING 1220 svchost.exe 2024-02-24 22:47:36.000000
0x50ed0d83e4d0_0TCPv6 :: 5840 :: 0 LISTENING 1220 svchost.exe 2024-02-24 22:47:36.000000
0x50ed0d8aa0f010_0TCPv4 192.168.182.139 49748 204.79.197.222 443 CLOSED 4780 SearchApp.exe 2024-02-24 22:50:42.000000
0x50ed0d8aa0f010_0TCPv6 :: 49748 :: 0 LISTENING 1220 svchost.exe 2024-02-24 22:47:36.000000
0x50ed0d857e400_0DPv4 ::::1 1900 * 0 7544 svchost.exe 2024-02-24 22:47:57.22:50:42.000000
0x50ed0d857e400_0DPv6 ::::1 1900 * 0 7544 svchost.exe 2024-02-24 22:47:57.22:50:42.000000
0x50ed0d8c52a0_0TCPv4 192.168.182.139 49719 23.222.237.202 443 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:48:47.000000
0x50ed0d8c52a0_0TCPv6 :: 49719 :: 0 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:48:47.000000
0x50ed0d9178a0_0TCPv4 192.168.182.139 49723 192.168.182.128 80 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:48:49.000000
0x50ed0d9178a0_0TCPv6 :: 49723 :: 80 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:48:49.000000
0x50ed0d91b60_0TCPv4 192.168.182.139 49812 192.168.182.128 80 CLOSE_WAIT 8300 msedge.exe 2024-02-24 22:52:40.000000
0x50ed0d91b60_0TCPv6 :: 49812 :: 80 CLOSE_WAIT 8300 msedge.exe 2024-02-24 22:52:40.000000
0x50ed0d9478a0_0TCPv4 192.168.182.139 49746 13.107.128.254 443 CLOSED 4780 SearchApp.exe 2024-02-24 22:50:47.000000
0x50ed0d9478a0_0TCPv6 :: 49746 :: 0 CLOSED_WAIT 4780 SearchApp.exe 2024-02-24 22:50:47.000000
0x50ed0d9508a0_0TCPv4 192.168.182.139 49744 23.222.237.203 443 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:50:39.000000
0x50ed0d9508a0_0TCPv6 :: 49744 :: 0 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:50:39.000000
0x50ed0d9d6a0_0TCPv4 192.168.182.139 49721 52.123.129.254 443 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:48:49.000000
0x50ed0d9d6a0_0TCPv6 :: 49721 :: 0 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:48:49.000000
0x50ed0df3a0_0TCPv4 192.168.182.139 49765 13.107.211.254 443 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:50:42.000000
0x50ed0df3a0_0TCPv6 :: 49765 :: 0 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:50:42.000000
0x50ed0d9fe3a90_0DPv6 fe00::185b:1837::9f7:bffd 59939 * 0 7544 svchost.exe 2024-02-24 22:47:57.000000
0x50ed0d9fe7140_0DPv6 fe00::185b:1837::9f7:bffd 1900 * 0 7544 svchost.exe 2024-02-24 22:47:57.000000
0x50edac57a0_0TCPv4 192.168.182.139 49712 152.199.55.200 443 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:48:06.000000
[...]
```

Using the windows.netscan plugin to identify the network connections and using grep to filter connection made to port 80

IP address = 192.168.182.128 Process that the owner used to access through port 80 is msedge.exe

Using the windows.pstree plugin

```
*** 8756 3196 powershell.exe 0xe50edab3f080 12 - 1 False 2024-02-24 22:48:02.000000 N/A
*** 9172 8756 systeminfo.exe 0xe50ed8f03340 0 - 1 False 2024-02-24 22:48:36.000000 PID of critical_updat is 1648
*** 8748 8756 conhost.exe 0xe50edac73340 3 - 1 False 2024-02-24 22:48:03.000000
*** 7960 3196 cmd.exe 0xe50edacdd080 1 - 1 False 2024-02-24 22:50:40.000000 N/A
*** 3384 7960 conhost.exe 0xe50edab37080 4 - 1 False 2024-02-24 22:50:40.000000 N/A
*** 1648 7960 critical_updat 0xe50ed94c1080 5 - 1 False 2024-02-24 22:51:50.000000 N/A
**** 1612 1648 updater.exe 0xe50edab3080 6 - 1 False 2024-02-24 22:51:50.000000 N/A
*** 6460 3196 FTK Imager.exe 0xe50edad09080 19 - 1 False 2024-02-24 22:52:18.000000 N/A
* 984 596 LogonUI.exe 0xe50ed7d44080 0 - 1 False 2024-02-24 22:47:36.000000 2024-02-24 22:47:54.000000
0564 6552 csrss.exe 0xe50ed9f020c0 10 - 2 False 2024-02-24 22:47:53.000000 N/A
6612 6552 winlogon.exe 0xe50ed9f130c0 4 - 2 False 2024-02-24 22:47:53.000000 N/A
* 6764 6612 LogonUI.exe 0xe50ed9ab3240 12 - 2 False 2024-02-24 22:47:53.000000 N/A
* 6748 6612 fontdrvhost.ex 0xe50ed9add140 6 - 2 False 2024-02-24 22:47:53.000000 N/A
* 6772 6612 dwm.exe 0xe50ed9ab5080 14 - 2 False 2024-02-24 22:47:53.000000 N/A
analyst@ip-10-10-19-07:~$
```

PID of critical\_updat is 1648

PPID of the updater.exe is equal to the critical\_updat PID hence the PID of the child process off the critical\_updat is 1612

Timestamp

## Task 5 Answers

**Answer the questions below**

Using the plugin "windows.netscan" can you identify the IP address that establish a connection on port 80?

192.168.182.128

✓ Correct Answer

Using the plugin "windows.netscan," can you identify the program (owner) used to access through port 80?

msedge.exe

✓ Correct Answer

Analyzing the process present on the dump, what is the PID of the child process of critical\_updat?

1612

✓ Correct Answer

What is the time stamp time for the process with the truncated name critical\_updat?

2024-02-24 22:51:50.000000

✓ Correct Answer

## Task 6 Finding Interesting Data

### Using the windows.filescan plugin

```
1 2 | analyst@ip-10-10-19-67: ~
root@kali: /home/kali/Downloads
analyst@ip-10-10-19-67: ~ vol -f memdump.mem windows.filescan | grep "critical_update.exe"
0xe50edaa3ac20.0\Users\user01\Documents\critical_update.exe 216
analyst@ip-10-10-19-67: ~
```

Path Found

Using the windows.filescan plugin to examine the files accessed that are stored in the memory dump, using grep to filter the output and only get results that match with critical\_update.exe

### Using the windows.mftscan.MFTScan plugin

```
1 2 | analyst@ip-10-10-19-67: ~
root@kali: /home/kali/Downloads
analyst@ip-10-10-19-67: ~ vol -f memdump.mem windows.mftscan.MFTScan | grep -i "important_document.pdf"
* 0xd389c5fbad280 FILE 111003anZing finFiled Archive FILE_NAME 2024-02-24 20:39:42.000000 2024-02-24 20:39:42.000000 2024-02-24 20:39:42.000000
analyst@ip-10-10-19-67: ~
```

Using the windows.mftscan MFTScan to extract data from the Master File Table. The MFT file contains info about every file in different volumes and piping the output to grep to match the important\_document.pdf string

### Using the windows.memmap plugin

Creating a memory dump of the memory region related to the process ID 1612 as discovered earlier is the process ID of the updater.exe process

```

1 2 | analyst@ip-10-10-19-67: ~
root@kali: /home/kali/Downloads
analyst@ip-10-10-19-67:~$ vol -f memdump.mem -o . windows.memmap --dump --pid 1612
Volatility 3 Framework 2.5.2

Progress: 100.00          PDB scanning finished
Virtual Physical      Size     Offset in File   File output
0x7ffe0000    0x13af000  0x1000  0x0        pid.1612.dmp
0x1626241000  0x17d16000  0x1000  0x1000    pid.1612.dmp
0x162624e000  0xd488000  0x1000  0x2000    pid.1612.dmp
0x162624f000  0x5b589000  0x1000  0x3000    pid.1612.dmp
0x16266ff000  0x77d7000  0x1000  0x4000    pid.1612.dmp
0x16269ff000  0x2c60a000  0x1000  0x5000    pid.1612.dmp
0x208c50e000  0x7b606000  0x1000  0x6000    pid.1612.dmp
0x208c50e1000 0x57ea1000  0x1000  0x7000    pid.1612.dmp
0x208c51d0000 0xe444000  0x1000  0x8000    pid.1612.dmp
0x208c51d4000 0x7355d000  0x1000  0x9000    pid.1612.dmp
0x208c51d5000 0x254ec000  0x1000  0xa000    pid.1612.dmp
0x208c51d6000 0x40000000  0x1000  0xb000    pid.1612.dmp

```

Using the strings command to extract the strings from the memory dump and using grep to match the strings with the string “http”

```

root@kali: /home/kali/Downloads
analyst@ip-10-10-19-67:~$ strings pid.1612.dmp | grep http
http://key.critical-update.com/encKEY.txt
http://key.critical-update.com/encKEY.txt
http://key.critical-update.com/encKEY.txt
http://key.critical-update.com/encKEY.txt
https://powerlift-frontdesk.acompli.net/api/incidents/
https://powerlift-frontdesk.acompli.net/api/incidents/

```

Using the command below to search for the string “http://key.critical-update.com/encKEY.txt” within the file `pid.1612.dmp`, it displays 10 lines before and 10 lines after each match, utilizing the `strings` command to extract readable text from the dump file.

```

root@kali: /home/kali/Downloads
analyst@ip-10-10-19-67:~$ strings pid.1612.dmp |grep -B 10 -A 10 "http://key.critical-update.com/encKEY.txt"
CommonProgramFiles(x86)=C:\Program Files (x86)\Common Files
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
h8H$
DriverData=C:\Windows\System32\Drivers\DriverData
8[G_
USERDOMAIN_ROAMINGPROFILE=DESKTOP-3NMNM0H
C:\Users\user01\Documents\updater.exe
WB0

```

```

@si/0/_dk_http://critical-update.com http://critical-update.com http://key.critical-update.com/encKEY.txt
HTTP/1.0 200 OK
Server: SimpleHTTP/0.6 Python/3.10.4
Date: Sat, 24 Feb 2024 22:52:40 GMT
Content-type: text/plain
Content-Length: 9
Last-Modified: Fri, 23 Feb 2024 22:56:51 GMT
192.168.182.128
cafebabe
ul1/0/_dk_https://microsoft.com https://microsoft.com https://edge.microsoft.com/entityextractiontemplates/api/v1/a
le&key=d414dd4f9db345fa8003e32adc81b362
1/0/_dk_https://critical-update.com https://critical-update.com https://key.critical-update.com/encKEY.txt/
-- 
<dependentAssembly>
  <assemblyIdentity
    type="win32"
    name="Microsoft.Windows.Common-Controls"
    version="6.0.0.0"
    processorArchitecture="*"
    publicKeyToken="6595b64144ccf1df"
    language="NDX(
http://key.critical-update.com/
http://key.critical-update.com/
http://key.critical-update.com/encKEY.txt
http://key.critical-update.com

```

Server Used By The Attacker Has Been Identified

## Task 6 Answers

Answer the questions below

Analyzing the "windows.filescan" output, what is the full path and name for critical\_update?

C:\Users\user01\Documents\critical\_update.exe

✓ Correct Answer

Analyzing the "windows.mftscan.MFTScan" what is the Timestamp for the created date of important\_document.pdf?

2024-02-24 20:39:42.000000

✓ Correct Answer

Analyzing the updater.exe memory output, can you observe the HTTP request and determine the server used by the attacker?

SimpleHTTP/0.6 Python/3.10.4

✓ Correct Answer

## Task 7 Conclusion

### Conclusion

In this scenario, we have put into practice the skills necessary to start digging into the world of memory forensics and practice with a tool like volatility, which is widely used among digital forensics professionals.

We learned how to gather information about the machine the dump belongs to, search for connections, enumerate and investigate processes, and examine the content for malicious patterns in a memory dump.

While all the information presented in this room can be used in real-life scenarios, it is possible to delve more deeply into the topic by examining more complex attacks. Some interesting material to get more into it can be the following:

- [Windows Forensics 2](#)
- [Linux Forensics](#)

- [iOS Forensics](#)
- [Windows Applications Forensics](#)

[Tryhackme Writeup](#)[Tryhackme Walkthrough](#)[Cybersecurity](#)[Forensics](#)[Blue Team](#)[Follow](#)

## Published in System Weakness

5.9K Followers · Last published 21 hours ago

System Weakness is a publication that specialises in publishing upcoming writers in cybersecurity and ethical hacking space. Our security experts write to make the cyber universe more secure, one vulnerability at a time.

[Follow](#)

## Written by Joseph Alan

218 Followers · 225 Following

Cloud Security Engineer | AWS Solutions Architect Professional | CompTIA Cysa+|AWS sysops admin with LAB | TryHackMe top 1%| HackTheBox Rank - Pro Hacker

## Responses (1)



What are your thoughts?

[Respond](#)



Kanwar Usama  
9 days ago

...

Super work done. Excellent. keep it up.



Reply

## More from Joseph Alan and System Weakness



Joseph Alan

## Threat Hunting Introduction

Task 1 Introduction

Sep 26, 2023 7 1



...



In System Weakness by AbhirupKonwar

## The best way to find private Bug-Hunting programs

Recon process to find private programs

Dec 25, 2024 234 7



In System Weakness by AbhirupKonwar

## Exposed Git Directory P1 Bug

Story of P1 Bug that turned out to be ?

Dec 11, 2024 312 3





Joseph Alan

## Expose TryHackMe Write-Up

Expose

Sep 24, 2023

5



...

See all from Joseph Alan

See all from System Weakness

## Recommended from Medium

```
d

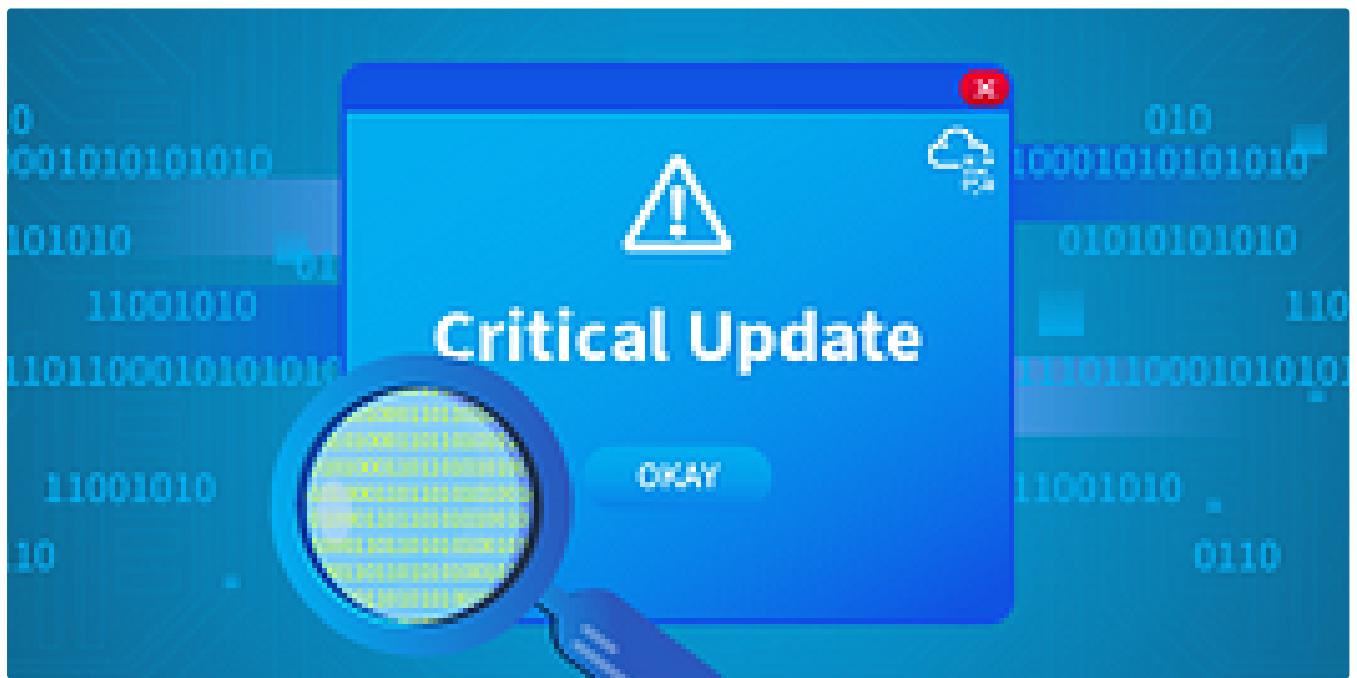
rd.img.old lib64 media opt root sbin srv tmp var vmlinuz.old
          lost+found mnt proc run snap sys usr vmlinuz
var/log
log# ls
cloud-init-output.log dpkg.log      kern.log    lxd       unattended-upgrades
cloud-init.log     fontconfig.log  landscape  syslog    wtmp
dist-upgrade      journal        lastlog    tallylog
log# cat auth.log | grep install
8-55 sudo: cybert : TTY=pts/0 ; PWD=/home/cybert ; USER=root ; COMMAND=/usr/bin/
8-55 sudo: cybert : TTY=pts/0 ; PWD=/home/cybert ; USER=root ; COMMAND=/usr/bin/
8-55 sudo: cybert : TTY=pts/0 ; PWD=/home/cybert ; USER=root ; COMMAND=/bin/chow
hare/dokuwiki/bin /usr/share/dokuwiki/doku.php /usr/share/dokuwiki/feed.php /usr/s
hare/dokuwiki/install.php /usr/share/dokuwiki/lib /usr/share/dokuwiki/vendor -R
log# █
```

 Dan Molina

## Disgruntled CTF Walkthrough

This is a great CTF on TryHackMe that can be accessed through this link here:  
<https://tryhackme.com/room/disgruntled>

Oct 22, 2024



 In T3CH by Axoloth

## TryHackMe | Critical | WriteUp

Acquire the basic skills to analyze a memory dump in a practical scenario.

Jul 21, 2024 104



...

## Lists



### Tech & Tools

22 stories · 380 saves



### Medium's Huge List of Publications Accepting Submissions

377 stories · 4345 saves



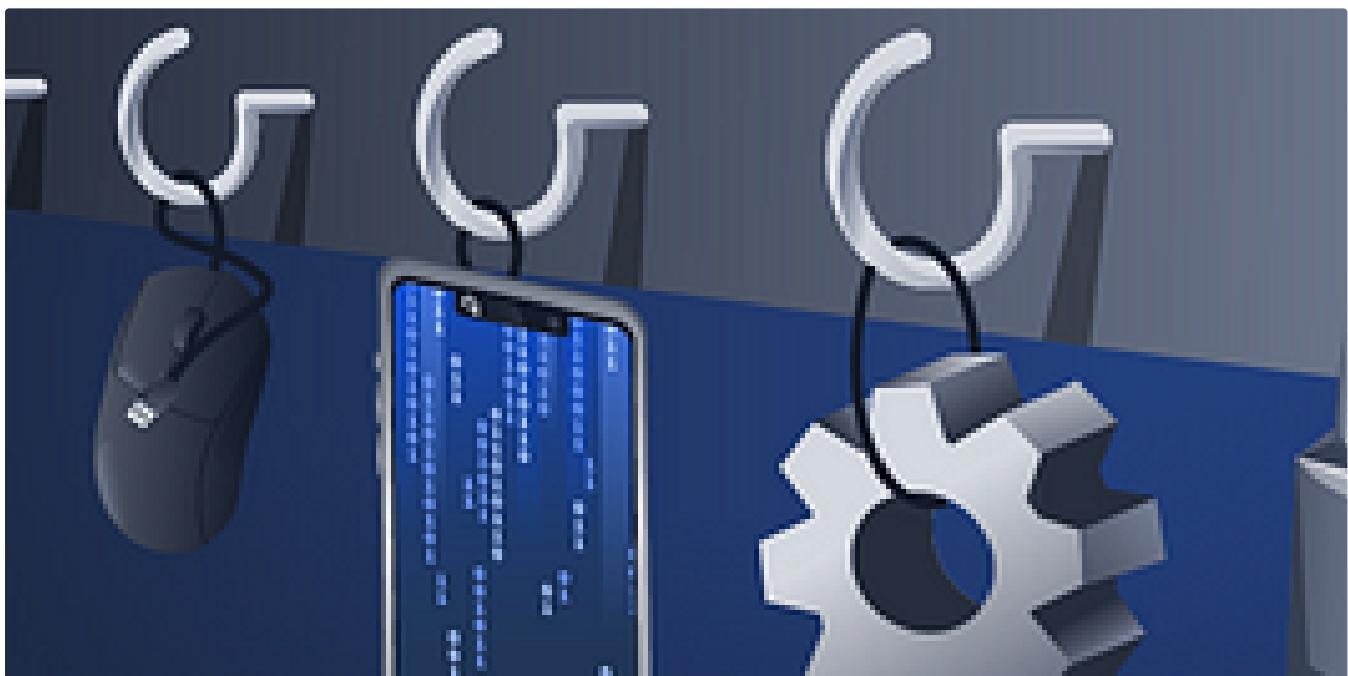
### Staff picks

796 stories · 1561 saves



### Natural Language Processing

1884 stories · 1529 saves



In T3CH by Axoloth

## TryHackMe | FlareVM: Arsenal of Tools| WriteUp

Learn the arsenal of investigative tools in FlareVM

Nov 28, 2024 50



...



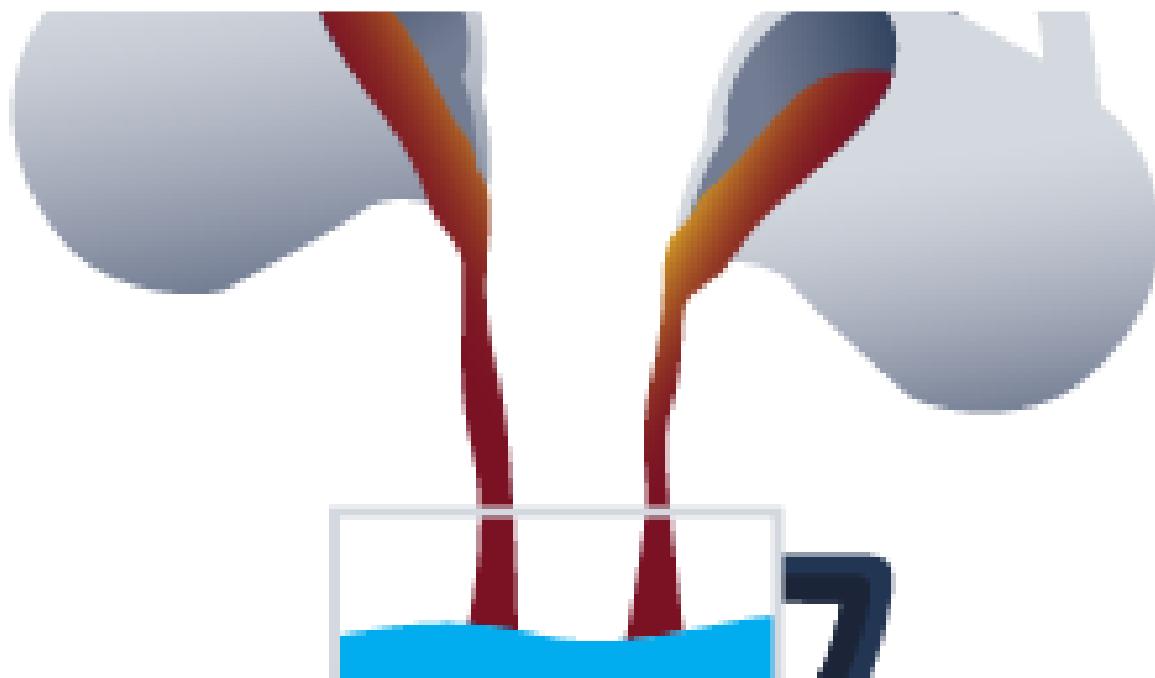
In T3CH by Axoloth

## TryHackMe | Training Impact on Teams | WriteUp

Discover the impact of training on teams and organisations

Nov 5, 2024

60



MAGESH

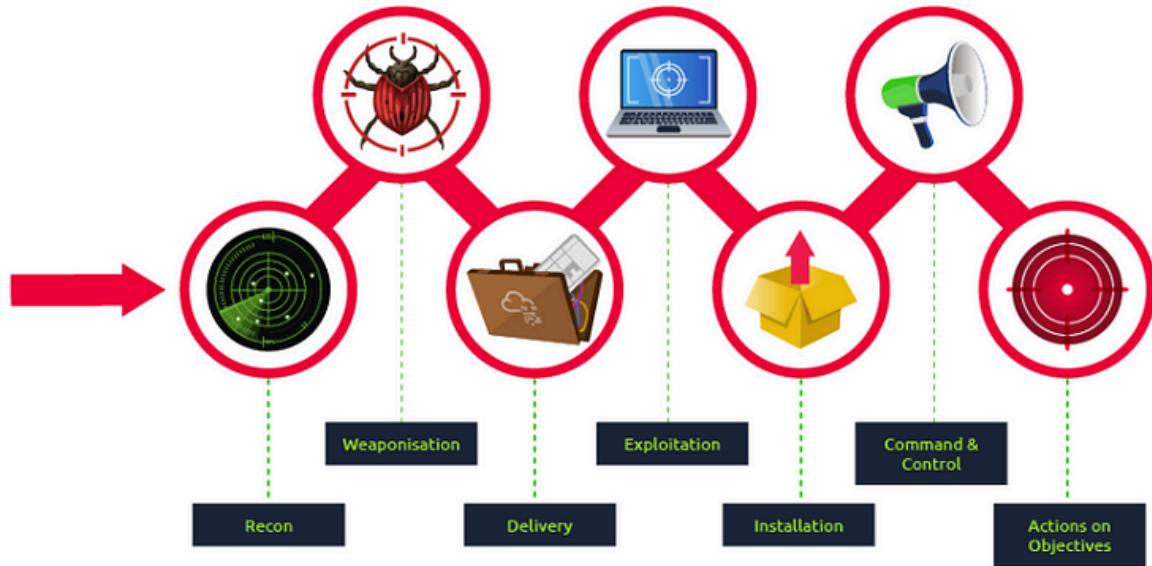
## Secret Recipe-Tryhackme Writeup

Perform Registry Forensics to Investigate a case.

Aug 21, 2024

2





 RosanaFSS

## TryHackMe: Threat Hunting With YARA, detailed Write-up

This room focuses on using YARA for threat hunting.

Nov 26, 2024



...

See more recommendations