

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



# [TryHackMe] Intro to Malware Analysis



Luigi Venditto · [Follow](#)

10 min read · Oct 5, 2023



Listen



Share

... More

I've decided to dive into the new SOC Level 2 room on THM, where the exciting world of Malware Analysis and threat hunting awaits. Malware has always intrigued me, though I've never had hands-on experience with it. Today, I invite you to learn alongside me as we explore this fascinating field.

This room is part of the SOC Level 1 learning path, and its goal is to introduce us to the basics of malware analysis and what to do when we encounter suspected malware.

Malware analysis is the art of dissecting malicious software, a vital skill in cybersecurity. Despite my limited experience, I'm excited to share what I learn. So, let's uncover the essentials of malware analysis together!

## Task 1

Welcome! In this section, we will provide an introduction to the room and give you an overview of what we will be learning. Once you've read this, please proceed to Task 2.

## Task 2

Here we have only one simple question to answer:

1. Which team uses malware analysis to look for IOCs and hunt for malware in a network?

To answer this, we need to refer to the notes provided in the task. In these notes, we learn that the term 'malware' is derived from 'MALicious' + 'softWARE.' Additionally, the importance of malware analysis in the security industry is highlighted, along with various roles involved in analysing malware, such as:

Security operations

Incident response

Threat hunters

Malware researchers

Threat research teams

Considering all this information, the answer is:

### **Threat hunt team**

Before we head to the next task, the notes also provide us with tips on how to handle suspected malware. These hints include using dedicated machines for malware analysis, storing samples in password-protected archives, extracting them only within a controlled environment, utilising isolated virtual machines that can be reset after analysis, and emphasising the importance of monitoring or closing internet connections. Additionally, it underlines the significance of cleaning the environment after each analysis to prevent contamination from previous malware executions.

### **Task 3**

In this task, we are introduced to two different types of analysis: static and dynamic analysis. Both forms of analysis are crucial for understanding how malware functions because they provide distinct pieces of information. As analysts, we can combine all the data to form a clear picture of the malware's operation.

The two questions in this task essentially tell us what each type does.

#### **1. Which technique is used for analysing malware without executing it?**

### **Static Analysis**

This is because static analysis focuses on checking for strings, examining PE headers for section information, and disassembling code. All of this can be done without executing it. However, some malware developers may use evasion techniques. Making static analysis limited.

## 2. Which technique is used for analysing malware by executing it and observing its behaviour in a controlled environment?

### Dynamic Analysis

While static analysis provides essential information, dynamic analysis becomes necessary, especially when a malware developer hides the malware properties through techniques like packing and obfuscation. Dynamic analysis involves running malware in a controlled environment, such as a VM or sandbox, to observe its behaviour. This controlled environment enables the effective identification of malware activity. However, some malware attempts to evade dynamic analysis by detecting the controlled environment and behaving innocuously in such instances. In such cases, advanced analysis techniques may be required, but this room does not go into this at this point.

### Task 4

Here we begin our static analysis, which is typically the first step when analysing new malware. The purpose of this step is to discover the malware's properties, providing an overview of what we are dealing with, including tasks such as identifying API calls or detecting packing. However, the primary objective of static analysis is to assess the malware's complexity and determine the level of effort required for further analysis.

In this room, we will be using the virtual machine from this task, and we will continue to use it in the next tasks as well, so remember to keep it open. This virtual machine already has REMnux installed, allowing you to utilise its commands to analyse some malware samples found in the sample folder.

As you read the notes, you will be introduced to various commands, including:

**file:** This command provides information about the file type.

**strings:** It lists all the strings present in a file, and using "> [new name]" will output the strings into a .txt file for easier reading.

**md5sum, sha1sum, and sha256sum:** These commands calculate the corresponding hashes of the file.

Now let's use this knowledge to answer the first question.

**1. In the attached VM, there is a sample named 'redline' in the Desktop/Samples directory. What is the md5sum of this sample?**

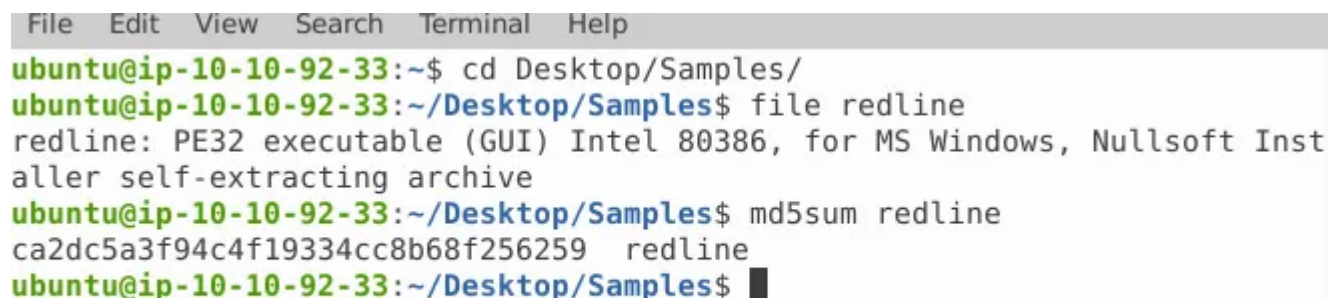
This task is straightforward. First, we need to ensure that we are in the directory containing the file we want to analyse. To do this, we use the following command:

```
cd /Desktop/Samples
```

This command takes us to the "Samples" directory, and from here, we can begin using REMnux commands to analyze the files we need. For this question, where we want the MD5 hash of "redline," we use the following command:

```
md5sum redline
```

This result will provide the MD5 hash value for the "redline" file.



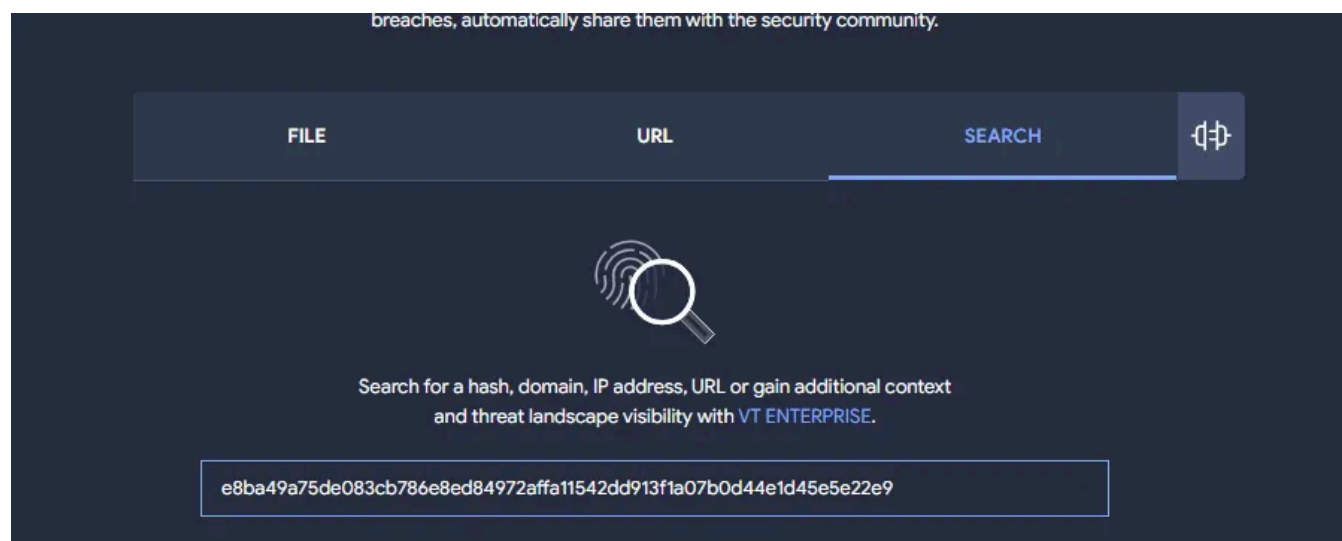
```
File Edit View Search Terminal Help
ubuntu@ip-10-10-92-33:~$ cd Desktop/Samples/
ubuntu@ip-10-10-92-33:~/Desktop/Samples$ file redline
redline: PE32 executable (GUI) Intel 80386, for MS Windows, Nullsoft Install
er self-extracting archive
ubuntu@ip-10-10-92-33:~/Desktop/Samples$ md5sum redline
ca2dc5a3f94c4f19334cc8b68f256259  redline
ubuntu@ip-10-10-92-33:~/Desktop/Samples$
```

You can use commands like **strings**, **sha1sum**, and **sha256sum** to explore the file further, uncovering details that might be of interest or helping you become more proficient in using these tools.

This task also introduces us to a very useful website and database, VirusTotal. Searching for a hash on VirusTotal can yield valuable insights into malware classification by security researchers. You can also do this on an Anti-Virus software that you may have.

**2. What is the creation time of this sample?**

To find out the creation time of this sample, start by copying the MD5 hash obtained from the previous question. Next, paste this hash into the search bar on VirusTotal.



Once you obtain the search result, navigate to the “details” section, and from there, go to the “history” tab. You should find all the necessary timestamps in this section.

History ⓘ	
Creation Time	2020-08-01 02:44:18 UTC
First Submission	2022-03-05 15:45:53 UTC
Last Submission	2023-05-30 05:18:39 UTC
Last Analysis	2023-09-29 04:49:25 UTC

## Task 5

Now that we have explored the basics of static analysis, this room will introduce us to more in-depth static analysis techniques by examining PE headers. Portable Executable (PE) headers provide crucial information about executable files in the Windows operating system. They encompass details about the file's structure, sections, imports, exports, and more. Analysing PE headers is valuable because they unveil insights into a file's functionality, dependencies, and potential malicious behaviour.

In the notes, there is an explanation of what to look for in the headers, and I recommend reading that for a better understanding. For this activity, we'll focus on the sections within a PE file. The most common sections include:

**.text:** This section typically contains CPU instructions executed when the PE file runs. It is marked as executable.

**.data:** This section contains global variables and other global data used by the PE file.

**.rsrc:** This section holds resources utilised by the PE file, such as images and icons.

Understanding these sections can provide valuable information during malware analysis.

To answer the questions, we will again use the REMnux tool, but this time we will utilise the “pecheck” command.

“pecheck” checks for irregularities or suspicious characteristics within the PE file’s structure, headers, or sections. This helps analysts identify potential anomalies or indicators of malicious behaviour in the file. It then displays information such as:

- **File Details:** Size, timestamps, and type.
- **PE Header Info:** Entry point, section count, and more.
- **Sections:** Names, sizes, characteristics, locations.
- **Imports:** Imported functions for functionality insights.
- **Exports:** Functions exported for other programs.
- **Checksum:** File integrity verification.
- **Entropy:** Level of compression or obfuscation.
- **Strings:** ASCII and Unicode strings for clues.
- **Headers Summary:** Concise overview of file headers.

As you can see, “pecheck” collects a lot of useful data for analysis.

1. **In the attached VM, there is a sample named ‘redline’ in the directory Desktop/Samples. What is the entropy of the .text section of this sample?**

In the terminal that we previously opened to find the hash of the “redline” file, we will now enter the following command:

*pecheck redline > pecheckred*

```
ubuntu@ip-10-10-163-184:~/Desktop/Samples$ pecheck redline >pecheckred
ubuntu@ip-10-10-163-184:~/Desktop/Samples$
```

Using the “> pecheckred” part is optional, but I personally prefer creating a .txt file to make it easier to read.

Now, if you’ve created the .txt file, open it and search for the “.text” section. Copy and paste the entropy value from that section into your answer.

```
1 pecheckred
PE check for 'redline':
Entropy: 7.999627 (Min=0.0, Max=8.0)
MD5 hash: ca2dc5a3f94c4f19334cc8b68f256259
SHA-1 hash: ce9943d9efc7d5f10cac4ab0b5aa48d62a063852
SHA-256 hash: e8ba49a75de083cb786e8ed84972affa11542dd913f1a07b0d44e1d45
SHA-512 hash: 8c774f64631342c2465d166cd4c374356c40c1cf6bae13b2e0b003ce6
.text entropy: 6.453919 (Min=0.0, Max=8.0)
.rdata entropy: 5.136718 (Min=0.0, Max=8.0)
.data entropy: 4.096809 (Min=0.0, Max=8.0)
.ndata entropy: 0.000000 (Min=0.0, Max=8.0)
.rsrc entropy: 4.209687 (Min=0.0, Max=8.0)
Dump Info:
-----DOS HEADER-----
```

**2. The sample named ‘redline’ has five sections. .text, .rdata, .data and .rsrc are four of them. What is the name of the fifth section?**

As we’ve read in the notes, “.text” is just one of the possible sections a file can have. Therefore, we should look for any other section that is not mentioned

we can see that the “.ndata” section is not mentioned, and that’s the answer.

```
SHA-512 hash: 8c774f64631342c2465d166cd4c374356c40c1cf6bae13b2e0b003ce6
.text entropy: 6.453919 (Min=0.0, Max=8.0)
.rdata entropy: 5.136718 (Min=0.0, Max=8.0)
.data entropy: 4.096809 (Min=0.0, Max=8.0)
.ndata entropy: 0.000000 (Min=0.0, Max=8.0)
.rsrc entropy: 4.209687 (Min=0.0, Max=8.0)
Dump Info:
```

**3. From which dll file does the sample named ‘redline’ import the RegOpenKeyExW function?**

This question highlights why I prefer saving the results as a .txt file. It allows us to use the Ctrl+F function to search for keywords. Since the “redline” file has a substantial number of strings, scrolling through all of them might cause us to miss

what we are looking for. However, by using Ctrl+F and searching for “regopen,” we can quickly find the answer to this final question.



There is also a GUI based pecheck result that can be opened with the command:

pe-tree [filename]

This will give the same result but presented in a fancier way.

## Task 6

Having explored some static analysis techniques, we are now delving into basic dynamic analysis. It's crucial to exercise caution during dynamic analysis because it involves executing potentially malicious files, which can harm your machine.

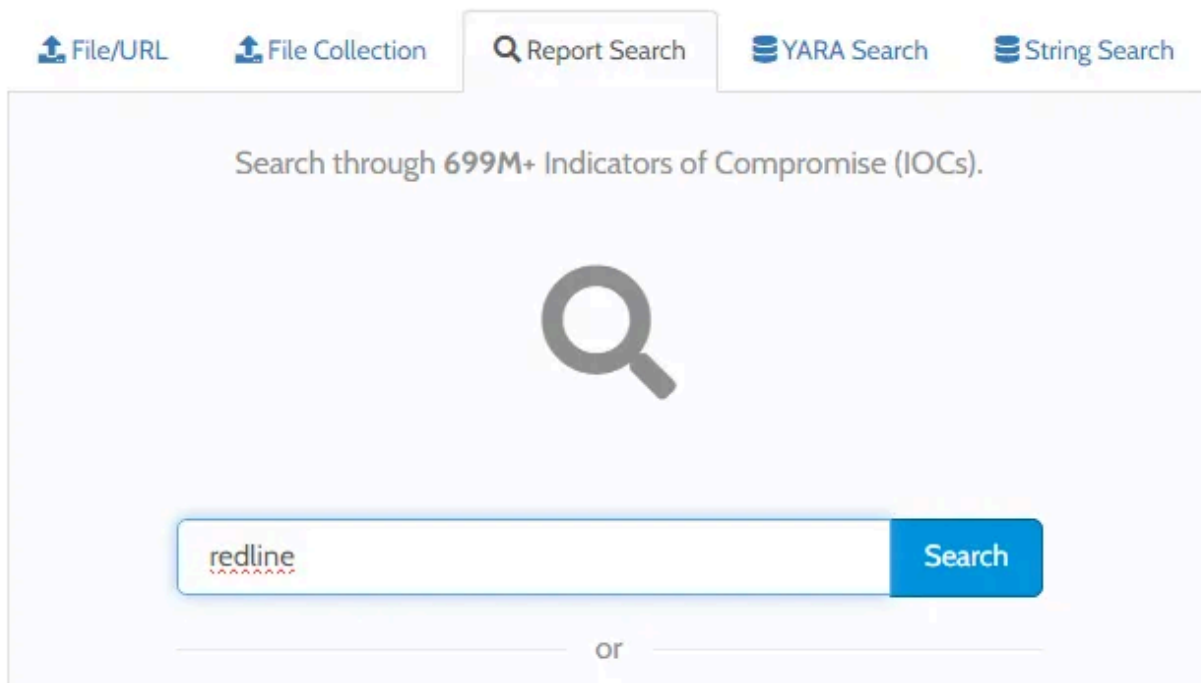
In today's task, we won't be executing any programs directly. Instead, we will be using the online platform Hybrid-Analysis.com. This platform offers a hybrid analysis service for files and URLs, combining both static and dynamic analysis techniques to assess the behaviour of potentially malicious content. As part of its process, it runs these files in a sandbox environment, which is an isolated setting that replicates the real target environment where an analyst can safely observe the sample's behaviour and learn more about it.

The task's notes provide additional details about sandboxes, including different types of sandboxes available. You can refer to these notes for a more comprehensive understanding of sandboxing and its various applications in malware analysis and cybersecurity.

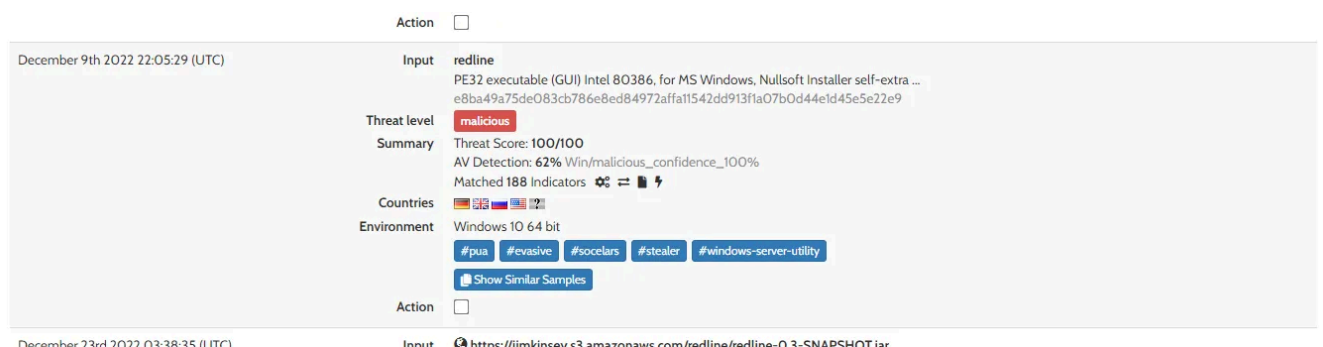
1. Check the hash of the sample 'redline' on Hybrid analysis and check out the report generated on 9 Dec 2022. Check the Incident Response section of the report. How many domains were contacted by the sample?

Go to the Hybrid-Analysis website and enter “redline” in the search bar.





From there, look for the entry dated 9th December in the timestamp column and open it.



Once the entry is open, search for network behaviour, and you should find a list of all the domains that it connects to.

	PE file has a section name known to be used by a packer/protector
	Tries to sleep for a long time (more than two minutes)
Exploit	Download executable files from web server
Spreading	Contains ability to enumerate volumes
Network Behavior	Contacts 17 domains and 10 hosts. <a href="#">View all details</a>

## 2. In the report mentioned above, a text file is accessed by the sample. What is the name of that text file?

The quickest way to do this is to use our good friend Ctrl+F and search for “.txt,” which will help us quickly locate what we need.



### Task 7

#### 1. Which of the techniques discussed above is used to bypass static analysis?

Malware authors employ various techniques to hinder an analyst's efforts. One common that they use is packing and obfuscation, which involves compressing, encrypting, or obfuscating malware content. These techniques make static analysis challenging, as packed malware does not reveal critical information during string searches. This evasion strategy makes it more difficult to do a static analysis.

**Answer : packing**

#### 2. Which technique discussed above is used to time out a sandbox?

Even dynamic analysis has its shortcomings. Malware developers employ several techniques to evade analysis, including:

- **Long Sleep Calls:** Delaying malicious activity to time out sandboxes, which typically run for a limited duration.
- **User Activity Detection:** Some malware waits for user interaction (e.g., mouse movement or keyboard input) before executing malicious actions, assuming no user activity in a sandbox. It can also detect patterns in mouse movements to bypass automated sandbox detection.

- **Footprinting User Activity:** Malware may check for user files or activity, such as MS Office history or internet browsing history. Minimal or absent activity may lead the malware to consider the system as a sandbox and terminate itself.
- **Detecting VMs:** Malware may identify virtual machines (VMs) based on artifacts left by VM software like VMWare or Virtualbox. Detecting a VM often leads to the termination of the malware as VMs are commonly associated with sandboxes.

These techniques aim to outsmart security analysts and automated environments, making it more challenging to detect and analyse malicious code effectively.

**Answer : long sleep calls**

This marks the end of our introduction to malware analysis. I hope you have enjoyed this journey, and now we have a fundamental understanding of what to expect during malware analysis. We explored different types of analysis, learned how to find strings and obtain hashes, and discovered various tools at our disposal. Additionally, it's important to remember the valuable resources available on the web that can help us in our malware analysis efforts.

[Malware Analysis](#)[Thm](#)[VirusTotal](#)[Hybrid Analysis](#)[Cybersecurity](#)[Follow](#)

## Written by Luigi Venditto

6 Followers · 2 Following

In this space, I'll be sharing my experiences as I navigate through various cybersecurity challenges and rooms.



## No responses yet

What are your thoughts?

Respond

## More from Luigi Venditto

```
desktop]
76" apache.log
[Jul/2023:12:35:02 +0000] "GET /contact.php HTTP/1.1" 200 123
x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.
[Jul/2023:12:34:54 +0000] "GET /contact.php HTTP/1.1" 200 789
x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.
[Jul/2023:12:34:47 +0000] "GET /index.php HTTP/1.1" 200 9876
64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.23
[Jul/2023:12:34:40 +0000] "GET /login.php HTTP/1.1" 200 9876
64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0.4638.11
[Jul/2023:12:34:33 +0000] "GET /about.php HTTP/1.1" 200 9876
64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.10
[Jul/2023:12:34:26 +0000] "GET /contact.php HTTP/1.1" 200 789
x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.
```

 Luigi Venditto

## [TryHackMe] Intro to Log Analysis

Introduction

Nov 1, 2023  8



```
all@WEBSRV-02:~$ sudo systemctl status rsyslog
log.service - System Logging Service
Loaded: loaded (/lib/systemd/system/rsyslog.service; vendor preset: enabled)
Active: active (running) since Sun 2023-10-08 13:12:40 UTC; 1min 45s ago
Main PID: 531 (rsyslogd)
CGroup: /system.slice/rsyslog.service
└─531 /usr/sbin/rsyslogd -n -iNONE
```

 Luigi Venditto

## [TryHackMe] Intro to Logs

Welcome to my new project: 'SOC Level 2' on THM. This path is designed to prepare you for Level 2 SOC roles and enhance your technical...

Oct 15, 2023  3

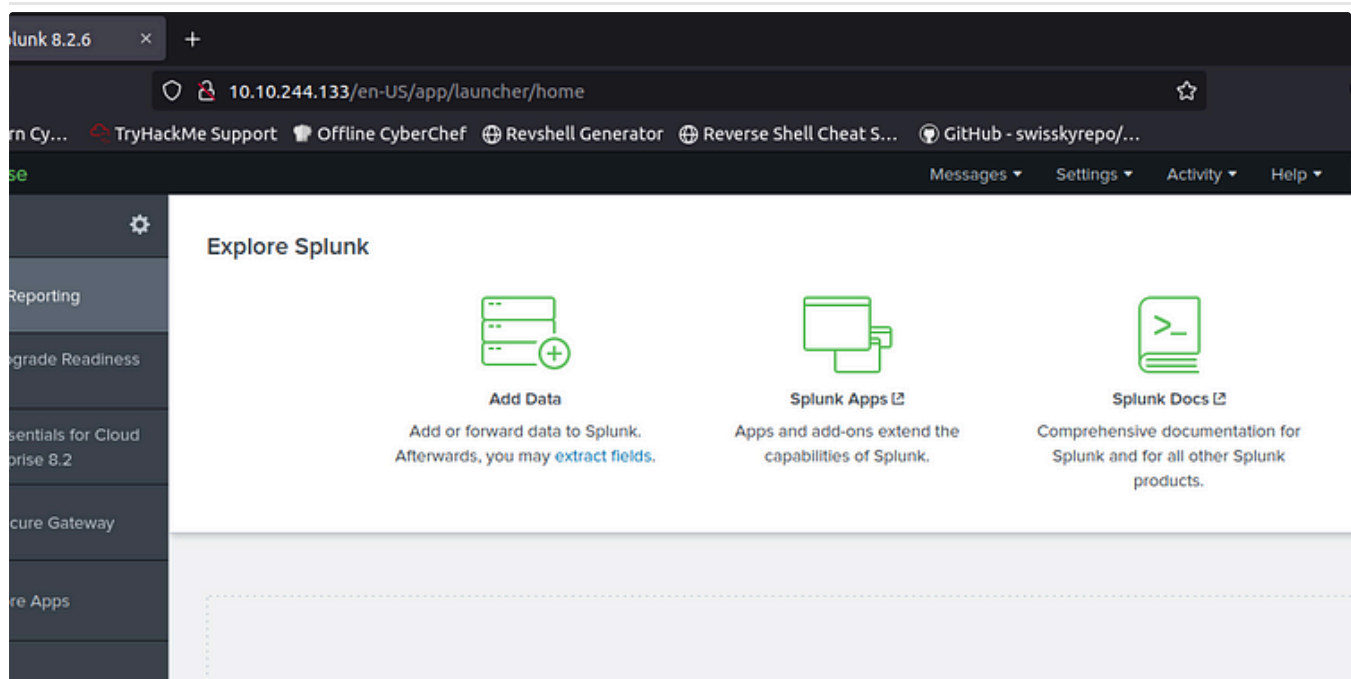


 Luigi Venditto

## [TryHackMe] Boogeyman 1

Introduction- "When you're afraid, close your eyes and count to five."

Nov 9, 2023 🖱 1

 Luigi Venditto

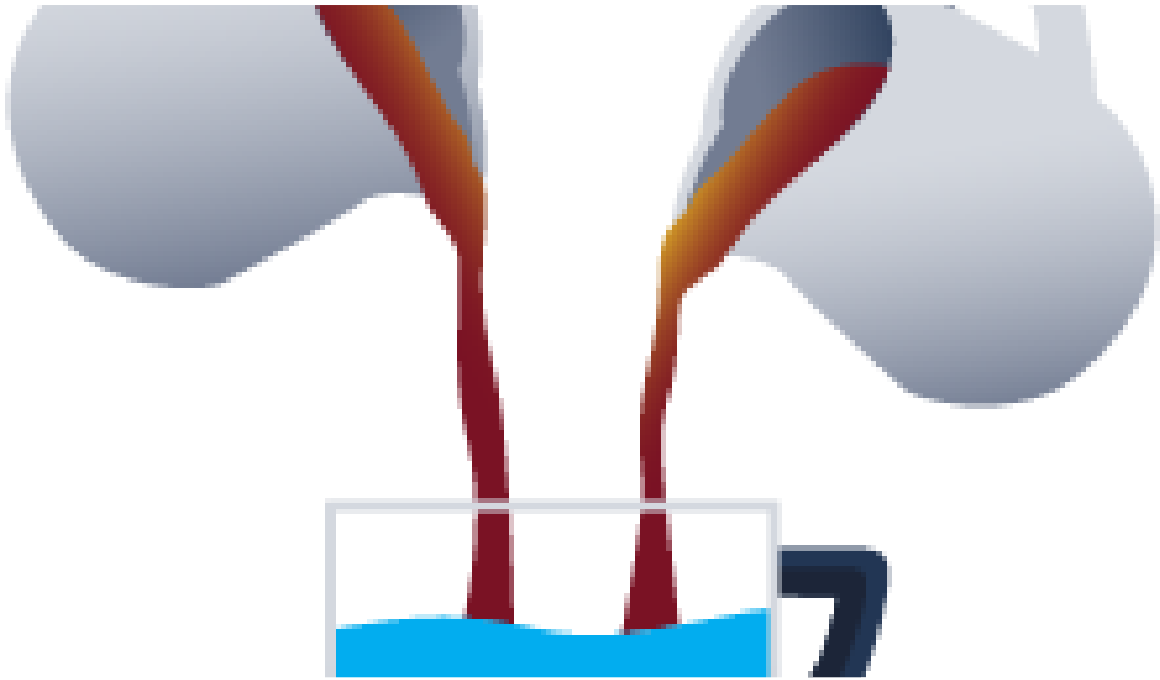
## [TryHackMe] Investigating with Splunk

Introduction

Oct 19, 2023 🖱 4

[See all from Luigi Venditto](#)

## Recommended from Medium



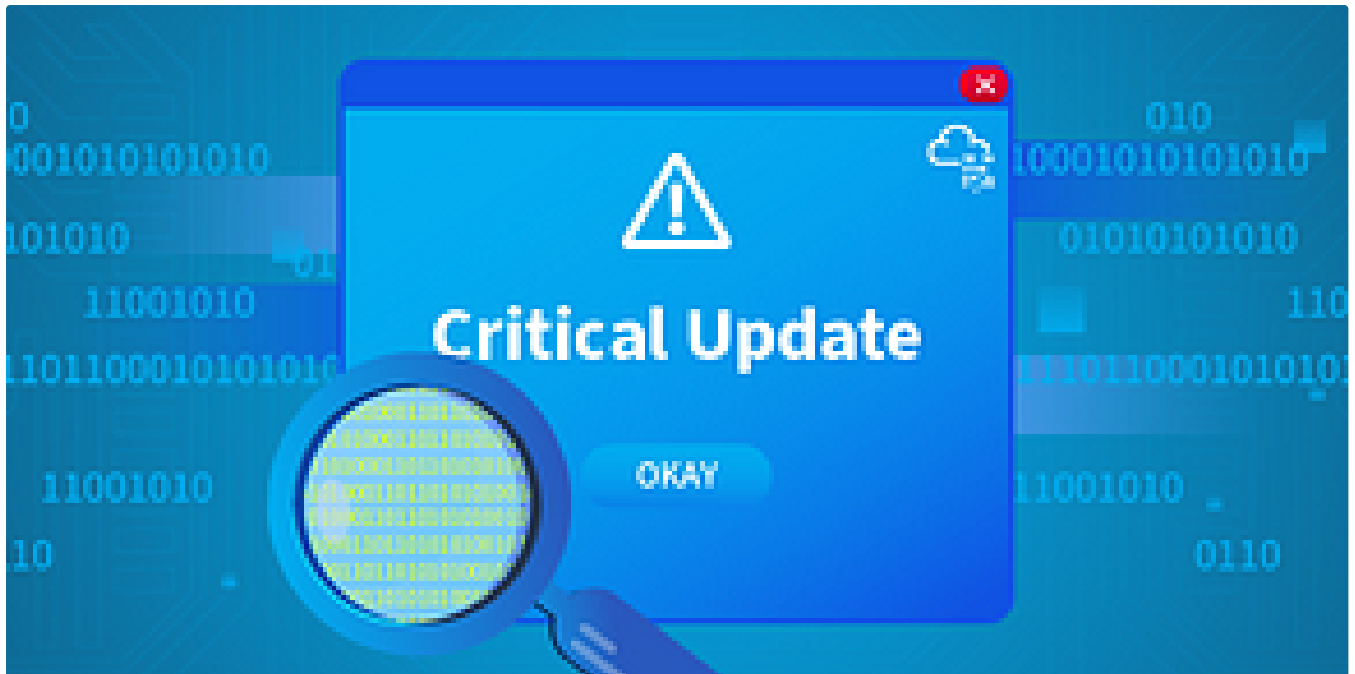
MAGESH

## Secret Recipe-Tryhackme Writeup

Perform Registry Forensics to Investigate a case.

Aug 21, 2024 🖐️ 2





 In T3CH by Axoloth

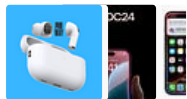
## TryHackMe | Critical | WriteUp

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

★ Jul 21, 2024 🖱️ 104

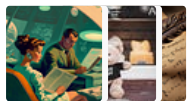


### Lists



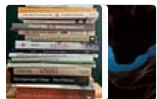
#### Tech & Tools

22 stories · 377 saves



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

377 stories · 4318 saves



#### Staff picks

793 stories · 1549 saves



#### Natural Language Processing

1883 stories · 1521 saves



[Open in app](#)**Medium** Search

Trnty

## TryHackMe | Introduction To Honeypots Walkthrough

A guided room covering the deployment of honeypots and analysis of botnet activities

★ Sep 7, 2024 🖱 10



Abhijeet Singh

### Advent of Cyber 2024 [Day 3] Even if I wanted to go, their vulnerabilities wouldn't allow it.

So, Let's Start with the Questions. I hope you already read the story and all the given instructions —

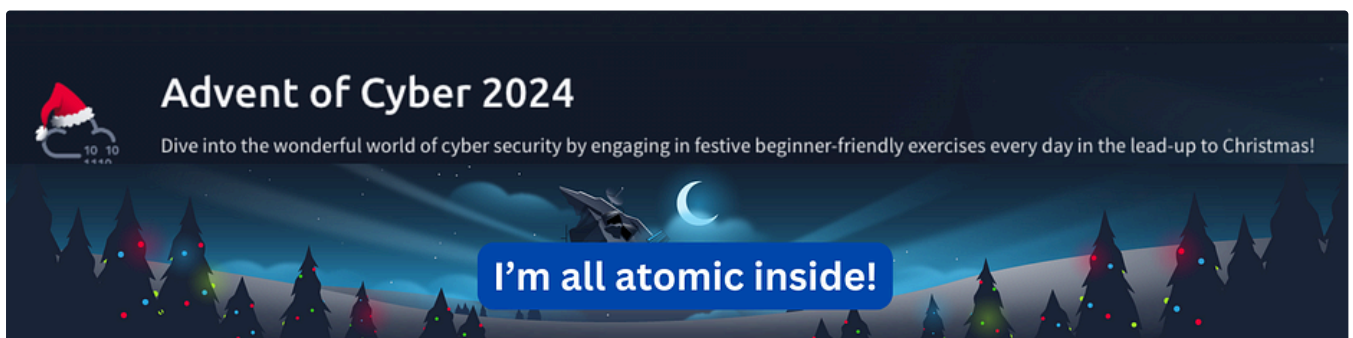
★ Dec 4, 2024 🖱 2

 In T3CH by Axoloth

## TryHackMe | FlareVM: Arsenal of Tools| WriteUp

Learn the arsenal of investigative tools in FlareVM

★ Nov 28, 2024 🖱 50



**Day 4**  
**Answers**

[cyberw1ng.medium.com](https://cyberw1ng.medium.com)

 In InfoSec Write-ups by Karthikeyan Nagaraj

# Advent of Cyber 2024 [ Day 4] Writeup with Answers | TryHackMe Walkthrough

I'm all atomic inside!

★ Dec 4, 2024 🖱 882 💬 1



See more recommendations