Room: BoogeyMAN 2 (https://tryhackme.com/room/boogeyman2)

Difficulty: Medium

Date: 14/08/2025

Status: Completed

Introduction

This is my walkthrough of the Boogeyman 2 challenge from TryHackMe, the second in a series of capstone challenges for the SOC Level 1 path.

The challenge is a multi-part DFIR (Digital Forensics & Incident Response) investigation focusing on a fictional threat actor called the Boogeyman.

Challenge Scenario

Quick Logistics LLC suffered an attack from the Boogeyman and improved its defenses. Unfortunately, the threat actor returned with updated tactics, techniques, and procedures (TTPs).

Maxine, an HR Specialist, received a malicious resume via email. This attachment compromised her workstation.

The SOC team flagged suspicious commands, prompting a full DFIR investigation.

Your task: analyze the artefacts, emails, and memory dump to unmask the Boogeyman’s methods.

Walkthrough & Questions

Q1: What email was used to send the phishing email?

Opened Resume – Application for Junior IT Analyst Role.eml - found From: field in email header.

Q2: What is the email of the victim employee?

Found in the To: field in the same header.

Q3: What is the name of the attached malicious document?

Searched for “attachment” in the `.eml` file → located the filename in Content-Disposition field.

Q4: What is the MD5 hash of the malicious attachment?

Downloaded attachment - ran:

md5sum Resume\_WesleyTaylor.doc

Q5: What URL is used to download the stage 2 payload?

Analyzed the document with olevba:

olevba Resume\_WesleyTaylor.doc

Extracted URL from VBA macros.

Q6: What is the name of the process that executed the stage 2 payload?

We analyzed the results in the previous question and discovered the process used to execute the payload.

Q7: What is the full file path of the malicious stage 2 payload?

We discovered the full path used to download and execute the file in Q5.

Q8: What is the PID of the process that executed the stage 2 payload?

Analyzed memory dump with Volatility 3:

vol -f WKSTN-2961.raw windows.pstree

Q9: What is the parent PID of the process that executed the stage 2 payload?

We can see the PPID in the output of pstree.

Q10: What URL is used to download the malicious binary executed by stage 2 payload?

Used Volatility’s process tree:

vol -f WKSTN-2961.raw windows.pstree | grep "PID"

Dumped child process memory with:

vol -f WKSTN-2961.raw windows.memmap --pid <CHILD-PID> --dump

strings pid.dmp | grep files.boogeymanisback.lol

Also searched entire memory dump:

strings WKSTN-2961.raw | grep files.boogeymanisback.lol

Q11: What is the PID of the malicious process used to establish the C2 connection?

Found via Volatility pstree / netscan plugins.

Q12: What is the full file path of the malicious process used to establish the C2 connection?

We already discovered where this executable was saved when we analyzed the javascript file.

However, we can confirm this by using the cmdline plugin to view command history.

Q13: What is the IP and port of the C2 connection?

Searched for active connections with netscan:

vol -f WKSTN-2961.raw windows.netscan | grep 6216

Output revealed IP:Port.

Q14: What is the full file path of the malicious email attachment based on memory dump?

Searched files with Volatility’s filescan:

vol -f WKSTN-2961.raw windows.filescan | grep Resume

Found Outlook temporary folder storing the malicious .doc.

Q15: What is the full command used to maintain persistence?

Looked for scheduled tasks with strings:

strings WKSTN-2961.raw | grep -i schtasks

Discovered malicious scheduled task command for persistence.

Conclusion

Mission accomplished!

We confirmed that the Boogeyman gained access via a malicious email attachment, downloaded a second-stage payload, established C2 communication, and maintained persistence with a scheduled task (schtasks).

This challenge was an excellent simulation of a real-world DFIR case study, combining email analysis, memory forensics, malware investigation, and persistence detection.

Tools & References

Olevba

Volatility 3

Volatility Command Reference

Microsoft Learn – schtasks

MITRE ATT&CK – Scheduled Task (T1053.005)