Jordan Knill
Final Assessment Report Submission

Case: One of us

1/15/2026

Executive Summary

This report details my investigation into a directory containing multiple suspicious executables. The objective was to develop and implement an automated scanning solution using the ClamAV API to efficiently identify malicious software. The investigation successfully transitioned from initial connectivity troubleshooting to the deployment of a functional bash script that identified a specific threat hiding among many benign files.

The process involved authenticating with the security API, handling SSL certificate verification issues, and refining script logic to accurately parse JSON responses. The outcome was the definitive identification of **file176.exe** as an infected file, while confirming the integrity of other scanned assets. This approach is a scalable methodology for rapid incident response and malware triage in workstation environments.

Findings and Analysis

Finding	Finding Details	Description
URL/API	https://clamav-ui.com/api/v1/scan	The endpoint used to perform signature-based scanning of the suspicious files.
Malicious File	file176.exe	This specific file was flagged by the scanner with a status of "infected": true. It was the only file in the batch to return this response.

Finding	Finding Details	Description
File Attribute	MD5: f48a8687e91fd9ef98cd1b7aaeeb2a4c	The unique cryptographic identifier or hash for the infected file file176.exe. can be used for identification and blacklisting.
File Attribute	sizeInBytes: 228550	The size of the malicious executable, consistent with the other files in the set, likely an attempt to blend in through uniformity with other files.
Process	scanner.sh	The custom-built automation script used to iterate through the directory and communicate with the ClamAV api.

Methodology

Tools and Technologies Used

The following tools were utilized to conduct the investigation and automate the scanning process.

- **Curl**: Used as the primary command-line tool for interacting with the REST API. It was essential for sending the JWT (JSON Web Token) for authorization and uploading the binary files using multipart/form-data.
- **Bash (Bourne Again SHell)**: Used to write the **scanner.sh** script, providing the logic for looping through the file directory.
- **Grep**: Utilized within the script to scrutinize the JSON responses and filter for the specific **"infected": true** string, allowing the script to ignore clean files and only alert on confirmed threats.
- **JSON API (ClamAV-UI)**: The technology used to perform the actual malware signature matching and return structured data regarding the file status.

Investigation Process

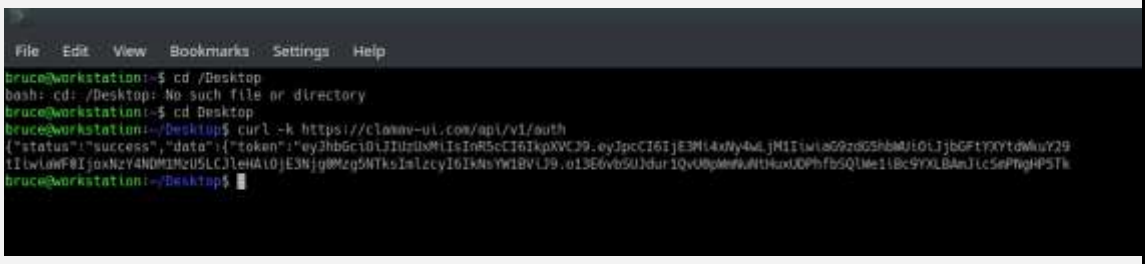
The investigation was conducted as follows to move from manual testing to full automation since doing this manually would take far too long.

1. **Initial Connectivity and Authentication**: I began by attempting to authenticate with the ClamAV API using curl. Initially attempts to authenticate and retrieve a token failed due to SSL certificate verification issues on the workstation.

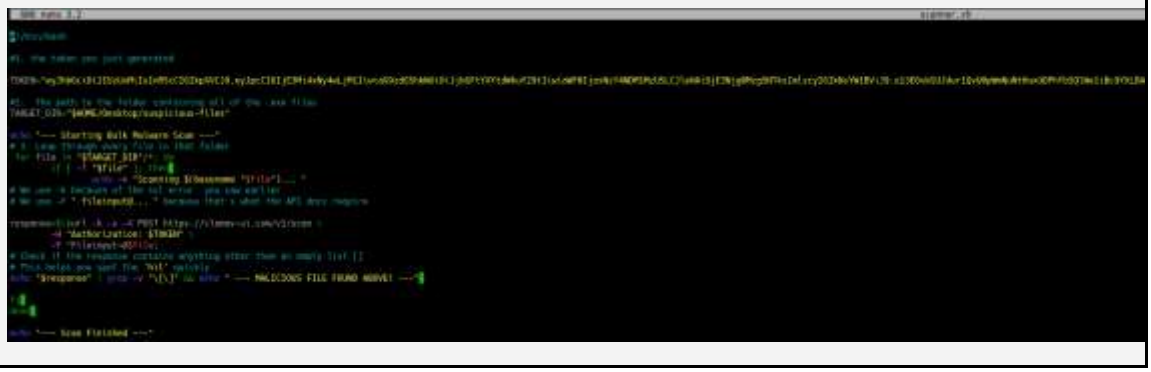
```
File  Edit  View  Bookmarks  Settings  Help
bruce@workstation:~$ curl https://clamav-ui.com/api/v1/auth
curl: (60) SSL certificate problem: unable to get local issuer certificate
More details here: https://curl.haxx.se/docs/sslcerts.html

curl failed to verify the legitimacy of the server and therefore could not
establish a secure connection to it. To learn more about this situation and
how to fix it, please visit the web page mentioned above.
bruce@workstation:~$
```

   so I switched to the folder containing suspected malicious files and resolved this by using the -k (insecure) flag to bypass local issuer certificate problems and successfully retrieved a JWT token.



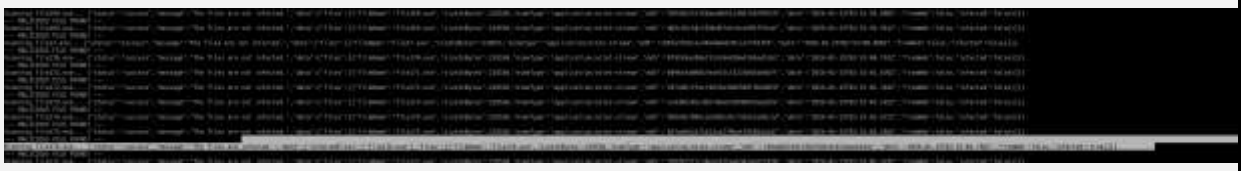
2. **Script Development**: I created **scanner.sh** to automate the process. I defined the TOKEN and the TARGET_DIR variables to point to the suspicious files on the Desktop.



3. **Refining Logic**: During the first run, I got a syntax error.

```
bruce@workstation:~/Desktop$ ./scanner.sh
--- Starting Bulk Malware Scan ---
./scanner.sh: line 28: unexpected EOF while looking for matching `"'
./scanner.sh: line 29: syntax error: unexpected end of file
bruce@workstation:~/Desktop$
```

4. During the 2nd run the script flagged every file because it was checking for a generic response rather than the specific "infected" value. I updated the script to use grep -q '"infected":true' to ensure only actual threats were reported.
5. **Execution and Identification**: I executed the final version of the script. It scanned through approximately 180 files, successfully identifying **file176.exe** as the malicious entity while reporting the rest as clean.



Recommendations

Based on the findings, I am proposing the following recommendations to mitigate the identified risks, secure the systems, and prevent future incidents.

1. Quarantine the malicious file immediately by moving file176.exe to an isolated directory and then delete it to prevent accidental execution.
2. Blacklist the Hash (**MD5 hash**: f48a8687e91fd9ef98cd1b7aaeeb2a4c) by adding it to the enterprise EDR (Endpoint Detection and Response) system to block this file.
3. Automate scanning by integrating the **scanner.sh** logic into a scheduled cron job to monitor the Downloads and Desktop folders for new threats automatically.
4. Update the local antivirus, ensure the local antivirus signatures are updated, as the file was found on the desktop, implying it may have bypassed this machines' defenses.