Noroff

CND 2025

SOC Log Pre-Processing Script

Tobias Axelson | tobaxe01194@stud.noroff.no
09/11/2025

Contents

Overview

This script automates the daily SOC pre-processing workflow. It securely downloads and validates configuration files, processes Apache logs, audits system integrity, generates reports, packages results and signs them for upload to a remote server defined by the operator. This method ensures traceability, data validation and operational security.

Task

The structure of the script is based on the provided tasks in the provided assignment, ranging from T1-24, while adhering to the K1-11 requirements.

Security

- HTTPS enforced for config and CTI downloads
- GPG signatures verified before usage
- Lockfile and cleanup function to prevent concurrent runs of the script
- SHA256 checksum ensure package integrity after transfer
- Audit logging at every step

Logging and output

- Main log: /opt/security/logs/socscript.log
- Reports: /opt/security/tmp/socscript-<date>/report.txt
- Validation: /opt/security/validation/validation.txt
- Uploads stored and verified on remote submission server

System requirements

- Operating system: Linux (tested on Ubuntu 24.04.3 LTS)
- Web Server: Apache2 (The script needs /var/log/apache2/access.log to function)
- Internet access: Required to download configuration and threat lists
- GPG Keys: soc-sign@dragur.no and soc-ingest@dragur.no
- Tools:
    - Curl

    - Gnupg
    - Rsync
    - Iptables
    - Net-tools
    - Findmnt
    - awk

Installation

To replicate this project, one would need to:

- Copy the script to /opt/security/
- Make the script executable
- Import SOC signing keys
- Ensure the Apache server is running
    - Folder structure used in this project:
        - /var/www/html/soc-hub.no/socscript/sigs
        - Sigs need to be included. -C parameter points to "soc-hub.no/socscript", and requires a subdirectory named sigs

How to run the script

Manual execution:

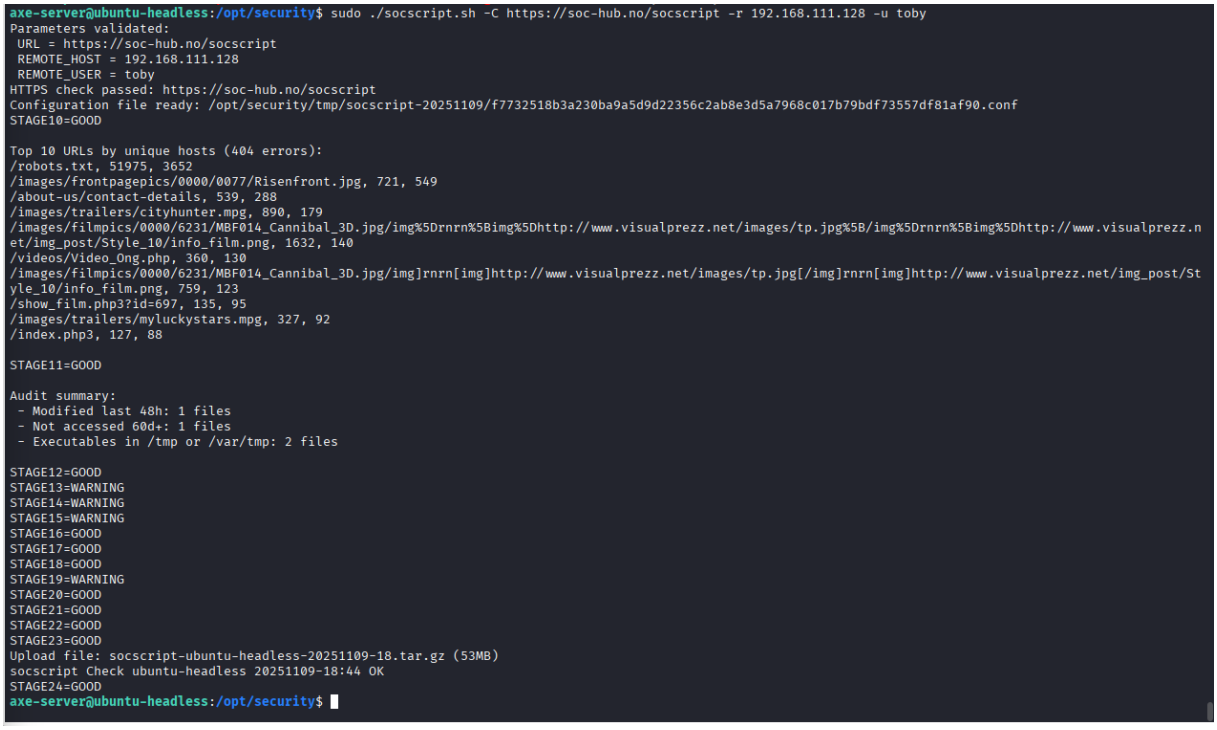
- sudo /opt/security/socscript-tobaxe01194.sh -C <URL> -r <REMOTE_HOST> -u <REMOTE_USER>
- Example: ./socscript-tobaxe01194.sh -C https://soc-hub.no/socscript -r 192.168.111.128 -u toby

Automated via system cron:

- sudo crontab -e
- Hit 1 for nano
- Add line:
    - 30 0 * * * opt/security/socscript-tobaxe01194.sh -C https://soc-hub.no/socscript -r 192.168.111.128 -u toby
- This ensures that the script is run at 02:30 AM daily

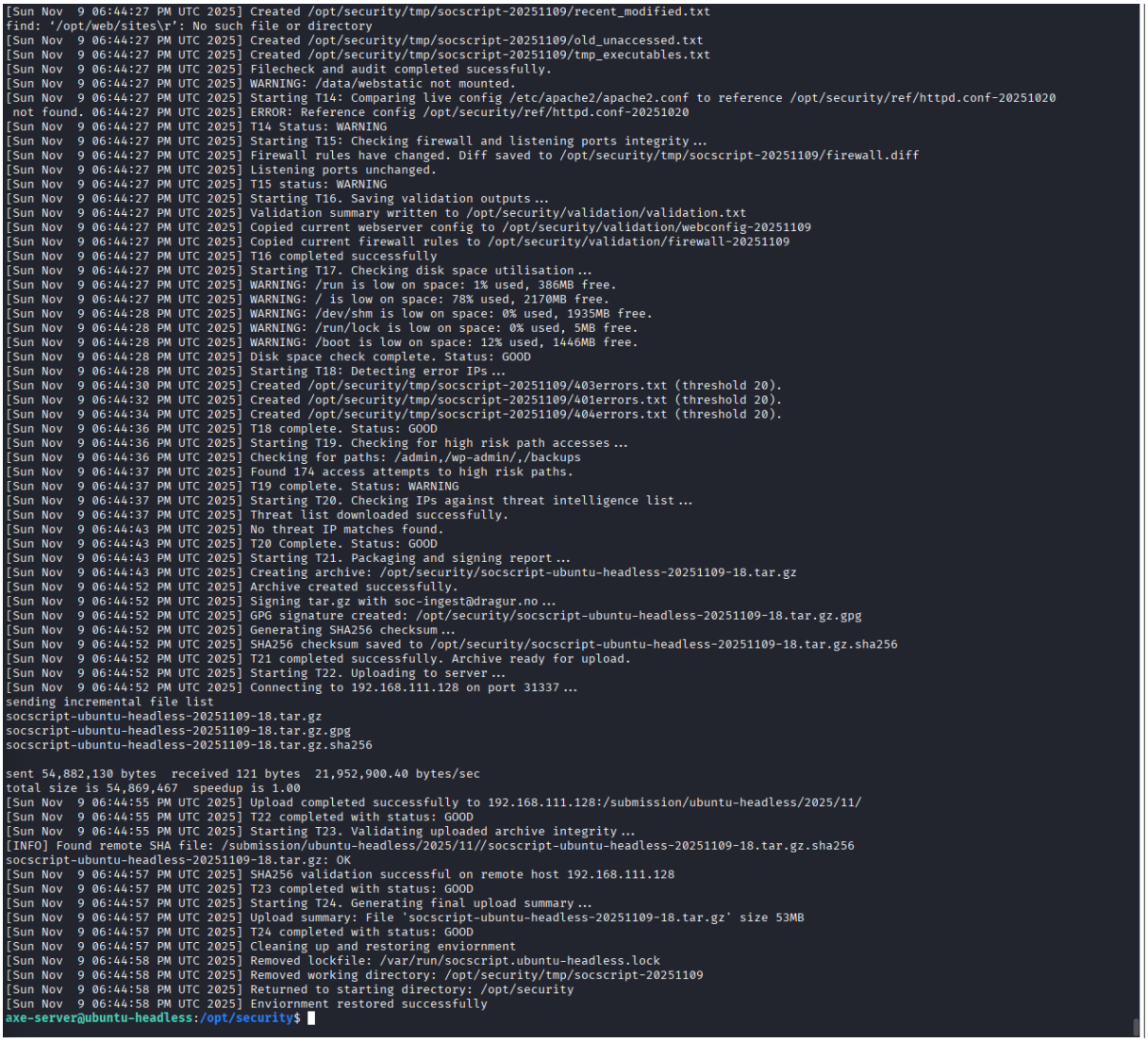
Testing evidence

Fig.1



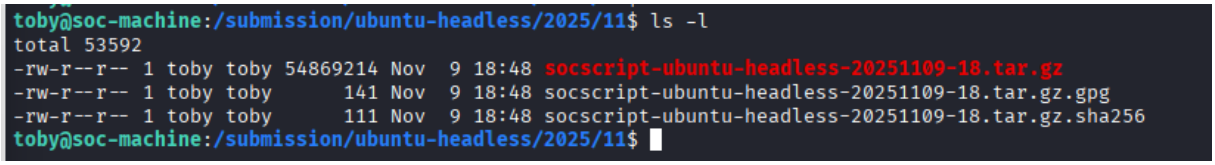
This screenshot show the script executed from command line. The script takes three arguments, -C <URL>, -r <REMOTE_HOST>, -u <REMOTE_USER>. Further the script outputs information about the process, and indicating if things are working as intended.

Fig.2



This screenshot shows output from the logfile, which the script appends various messages depending on how the script executes.

Fig.3



This final figure shows the uploaded files at the remote location.

Potential Problems

Missing or incorrect configuration files

- If (hostname).conf or default.conf cannot be downloaded the script will terminate
- This happens if user supply wrong URL or without https:// prefix

Missing GPG keys

- The script requires SPC signing keys (soc-sign@dragur.no and soc-ingest@dragur.no)
- If keys are not imported beforehand, signature validation and packaging will fail
- Use gpg –import <key.pub>

Permission or ownership issues

- The script writes to /opt/security, /var/run and reads from /var/log/apache2
- Running as non-root will cause permission errors
- Always run as sudo

Network connectivity issues

- The script relies on internet connectivity to download configuration files, threat lists and remote upload

Missing dependencies

- The script relies on curl, gpg, rsync, iptables, findmnt and awk.
- Should already be installed

Invalid paramerer usage

- The script accepts 6 arguments -C, -r, -u followed by a value
- Not using those flags and values will cause error

Log rotation timing

- If log rotation has not occurred, it may reuse access.log instead of access.log.1