FIRMWARE TRIAGE AND VULNERABILITY ANALYSIS

AUTOMATED EXTRACTION AND SCANNING OF EMBEDDED FIRMWARE

TEAM BETTERTEAM – JOHN FREEBORN, LEYTON MEADOWS, SPENCER CHRISTENSEN

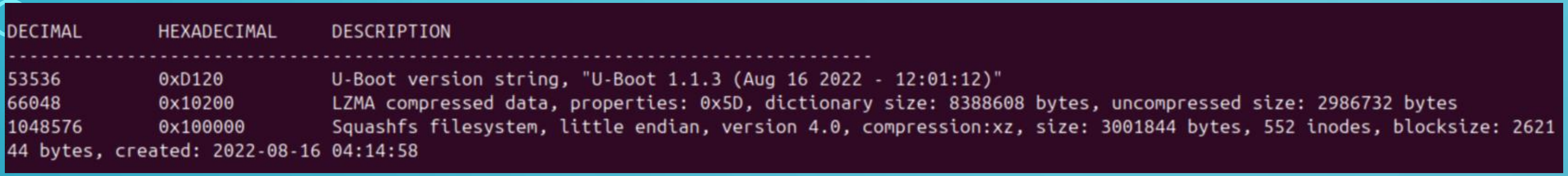
INTRODUCTION

- Client: ThanosTech LLC
- Scope: Static analysis of firmware samples
- Targets:
  - TP-Link TL-WR841N Router
  - D-Link DCS-8000LH Camera
  - Vare-metal STM Microcontroller ELF
- Objective: Identify hard-coded credentials, insecure configs, and embedded secrets

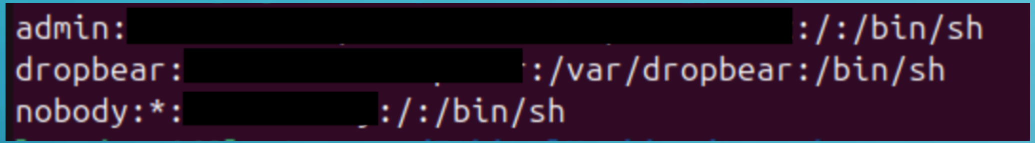
STRUCTURED ANALYSIS WORKFLOW

- OSINT Reconnaissance – Collected public data, CVEs, etc.
- Initial Recon – Determined image type of architecture (via file and binwalk)
- Extraction & Exploration – Unpacked filesystems using binwalk –eM
- Enumeration – Mapped services, startup scripts, and BusyBox versions
- Vulnerability Discovery – Used ripgrep and strings to locate secrets
- Disassembly – objdump & readelf to analyze ELF binaries
- Automation – Custom script for repeatable triage

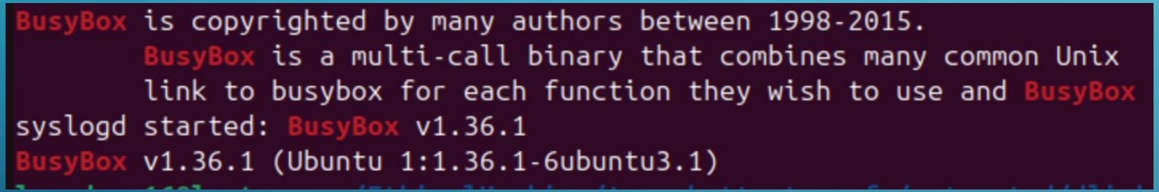
MAJOR VULNERABILITIES IDENTIFIED



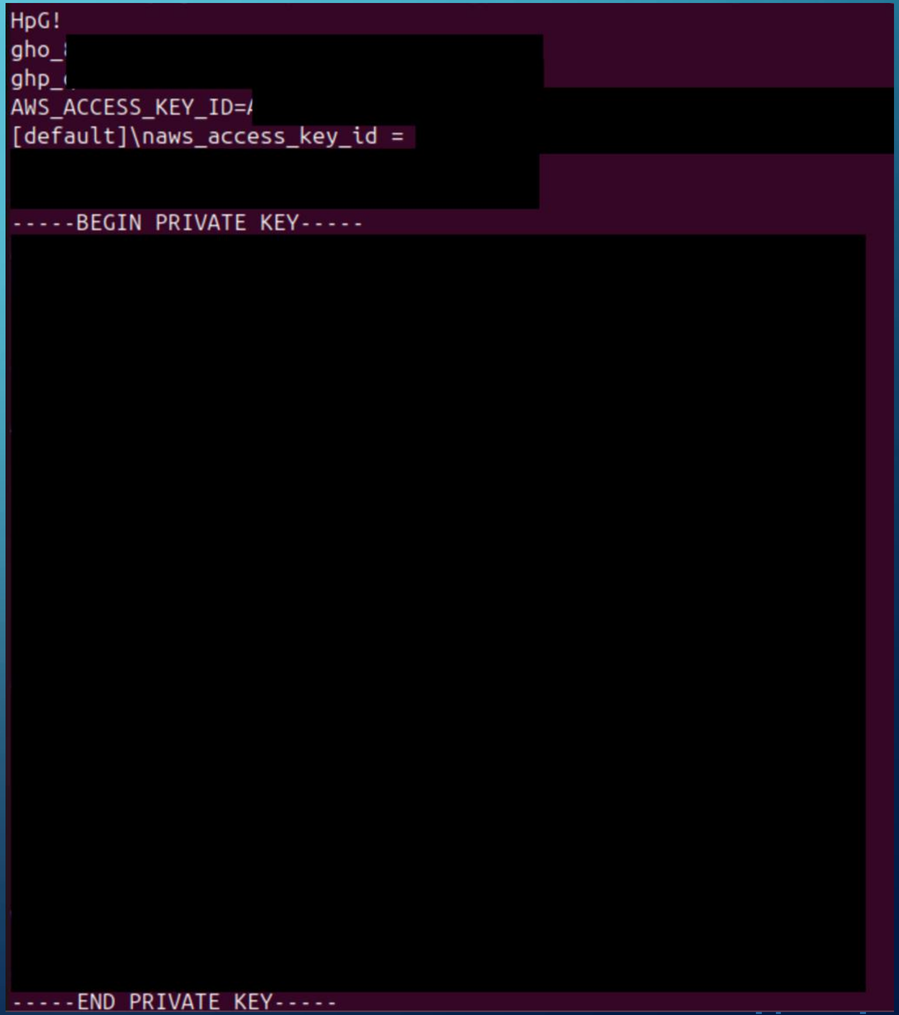
```
DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
53536         0xD120          U-Boot version string, "U-Boot 1.1.3 (Aug 16 2022 - 12:01:12)"
66048         0x10200         LZMA compressed data, properties: 0x5D, dictionary size: 8388608 bytes, uncompressed size: 2986732 bytes
1048576       0x100000        Squashfs filesystem, little endian, version 4.0, compression:xz, size: 3001844 bytes, 552 inodes, blocksize: 2621
44 bytes, created: 2022-08-16 04:14:58
```



```
admin:[redacted]:/:/bin/sh
dropbear:[redacted]:/var/dropbear:/bin/sh
nobody:*:[redacted]:/:/bin/sh
```



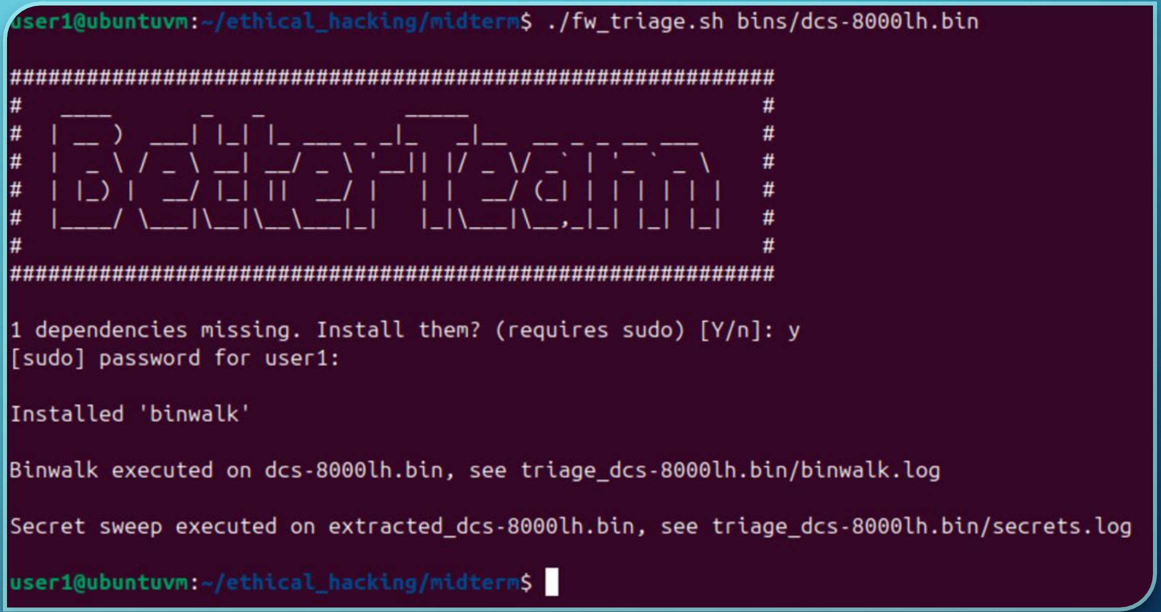
```
BusyBox is copyrighted by many authors between 1998-2015.
        BusyBox is a multi-call binary that combines many common Unix
        link to busybox for each function they wish to use and BusyBox
syslogd started: BusyBox v1.36.1
BusyBox v1.36.1 (Ubuntu 1:1.36.1-6ubuntu3.1)
```



```
HpG!
gho_[redacted]
ghp_[redacted]
AWS_ACCESS_KEY_ID=[redacted]
[default]\naws_access_key_id = [redacted]
-----BEGIN PRIVATE KEY-----
[redacted]
-----END PRIVATE KEY-----
```

AUTOMATION SCRIPT: *FW_TRIAGE.SH*

- Automates extraction and scanning across multiple images
- Detects file type, the runs:
  - Binwalk
  - Strings
  - Readelf
- Searches for keywords like “password”, “key”, “AWS”, “token”
- Saves logs automatically



```
user1@ubuntuvm:~/ethical_hacking/midterm$ ./fw_triage.sh bins/dcs-8000lh.bin

#############################################################
#   ____       _   _           _____                        #
#  | __ )  ___| |_| |_ ___ _ _|_   _|__  __ _ _ __ ___      #
#  |  _ \ / _ \ __| __/ _ \ '__|| |/ _ \/ _` | '_ ` _ \     #
#  | |_) |  __/ |_| ||  __/ |   | |  __/ (_| | | | | | |    #
#  |____/ \___|\__|\__\___|_|   |_|\___|\__,_|_| |_| |_|    #
#                                                           #
#############################################################

1 dependencies missing. Install them? (requires sudo) [Y/n]: y
[sudo] password for user1:

Installed 'binwalk'

Binwalk executed on dcs-8000lh.bin, see triage_dcs-8000lh.bin/binwalk.log

Secret sweep executed on extracted_dcs-8000lh.bin, see triage_dcs-8000lh.bin/secrets.log

user1@ubuntuvm:~/ethical_hacking/midterm$ 
```

MAJOR VULNERABILITY POTENTIAL FIXES

- Remove Hardcoded Passwords – Require unique credentials on first boot and disable Telnet
- Protect Keys & Secrets - Strip embedded keys; store device-unique keys in secure hardware
- Enable Secure Boot – Use verified boot (U-Boot > 2020.01) and validate firmware with digital signatures
- Update Vulnerable Components – Upgrade BusyBox and apply routine CVE patching
- Clean Build Artifacts – Remove debug paths, test files, and restrict access to sensitive directories
- Integrate Security in Development – add static-analysis and secret scanning to the CI/CD pipeline

FUTURE WORK & QUESTIONS

- Extend *fw_triage.sh* for CVE mapping
- Integrate secret scanning into CI/CD
- Continue firmware reverse engineering