Egyptian Informatics Journal 24 (2023) 100409

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

Forensic analysis and security assessment of IoT camera firmware for smart homes

Akashdeep Bhardwaj [a], Keshav Kaushik [a], Salil Bharany [b,*], SeongKi Kim [c,*]

[a] University of Petroleum and Energy Studies, Dehradun, India
[b] Department of Computer Science and Engineering, Lovely Professional University, Phagwara, Punjab 144402, India
[c] National Centre of Excellence in Software, Sangmyung University, Seoul 03016, Korea

ARTICLE INFO

Keywords:
IoT Security
IoT Camera
IoT Firmware
Firmware Analysis
Security Assessment

ABSTRACT

Ease of flexibility, convenience, and smartness have made the Internet of Things (IoT) the industry, and user favorite device has recently piqued the industry's interest. With the widespread use of this technology, an unprecedented number of IoT endpoints, including devices in our homes, are now connected to external networks. Firmware of this type is typically heterogeneous and closed source. At the firmware level, it is harder to detect and evaluate security vulnerabilities since their effects are faster and larger. In recent years, several similarity-based firmware security detection techniques and tools have become research hotspots. Security concerns and attacks have risen to the top of the list for IoT security equipment, particularly IoT cameras. Firmware has historically been one of the most neglected areas of device security, leaving it exposed to malicious actors. This research proposes a unique twelve-step process to perform firmware analysis and security assessment of Smart IoT-based Camera devices.

1. Introduction

The number of IoT devices has been growing because of the recent growth in smart factories and smart cities, as well as the fourth industrial revolution, and as a result, IoT device security has become a major concern. Existing security methods are difficult to deploy due to variables such as the poor performance of IoT nodes, and even if secure firmware is given, security difficulties may arise because of intrusions such as man-in-the-middle and roll-back assaults. The IoT market has grown significantly in recent years, attracting the attention of security researchers. The enormous range of IoT device hardware and software and their hardware designs, on the other hand, makes research difficult. Device firmware pictures, embedded firmware source code, and even development documents are not available from manufacturers of lightweight IoT devices. Consequently, traditional dynamic and static analysis methods are limited. According to a Microsoft IoT Security survey [1], 83 percent of 1,000 firms polled had at least one firmware attack in the last two years. According to Gartner [2], 70 percent of businesses without a firmware updating plan will experience a breach related to firmware vulnerability by the end of 2022. The National Institute of Standards and Technology's (NIST) National Vulnerability Database [3] reveals a more than fivefold increase in firmware assaults in the previous four years.

The IoT has its own set of firmware difficulties. These gadgets appear to function mystically, but they are the result of a collection of numerous different microprocessors within each one. These specialized microprocessors serve a variety of purposes: some are for graphics, others offer a fast and connected response to your input, and others support AI for intelligent capabilities and other tasks. Each has its firmware layer (software written specifically for each piece of hardware) and may run on multiple operating systems. The firmware allows a device to fulfill its intended duties and ensures the proper operation of a range of physical components. The device kernel, the file system, which stores specific files essential for device performance, and the bootloader, which is responsible for initializing important hardware components and assigning the necessary resources, are examples of such components.

The firmware also keeps sensitive data in memory, such as encryption keys. Firmware updates on linked devices are more difficult than software updates on desktop computers and laptops. Connected devices are made up of several manufacturers' products, many of which employ vendor-specific hardware and firmware. While manual firmware upgrades are possible, they are time-consuming and inconvenient, which

* Corresponding authors.
E-mail addresses: bhrdwh@yahoo.com (A. Bhardwaj), officialkeshavkaushik@gmail.com (K. Kaushik), salil.bharany@gmail.com (S. Bharany), skkim9226@gmail.com (S. Kim).

https://doi.org/10.1016/j.eij.2023.100409
Received 4 June 2023; Received in revised form 9 September 2023; Accepted 28 September 2023
Available online 20 October 2023

has led to firmware becoming a generally unprotected attack vector. Hackers use firmware to get into networks, gain access to other systems, compromise data, and even take over a device. By injecting malware into a device's firmware, bad actors can avoid antivirus inspections. Firmware security is a must-have for device makers, vendors, and system integrators.

2. Literature survey

Son et al. (2019) [4] secured the stability of the firmware, the authors suggested a novel firmware administration architecture based on Blockchain and the Inter-Planetary File System. However, because IoT device version control and URL authenticity cannot be assured, IoT devices can now get firmware updates and updates over Blockchain networks. The suggested approach allows for secure firmware transmission and updates, and the security level of IoT devices is projected to improve. According to the findings, secure firmware delivery and updates are conceivable, and IoT device security standards are predicted to improve.

Remote firmware upgrades on IoT devices result in periodic device failure and performance deterioration. To upload the updated code with updates or corrections and re-install the gadget on the field, the gadget needed physical access to a computer. Companies, on the other hand, will find this approach to be extremely inefficient and unscalable. Over-the-air firmware updates, on the other hand, offer a unique technique to upgrade linked devices without interfering with them, remotely and consistently. Thakur et al. (2019) [5] developed a scheme for updating the firmware of various IoT devices over the internet.

The identification of harmful code in firmware, susceptibility mining, backdoor finding, and copyright protection can all benefit from similarity and homology analyses of firmware codes in IoT terminals. To examine the topic of code resemblance and homology analysis, Zhu et al. (2020) [6] concentrated on code categorization and qualitative characterization of code aspects. The authors took an information-centric approach to firmware development, concentrating on firmware stability, anti-variability, and heritability. By analyzing the gene length between the codes, this study demonstrated a security detection solution for IoT terminal firmware. The results of the experiments reveal that this technique has a decent search pairing effect and has certain benefits over previous similarity theory-based firmware security investigative techniques.

Zhu et al. (2019) [7] introduced a unique dynamic analysis approach for detecting memory corruption issues in IoT device firmware packages that are lightweight. The main aim was to fuzz the binary program code while dynamically running them using symbolic execution. The authors created a prototype, and the findings revealed that the suggested framework could perform the Fuzzing test in an average of 40 s. In all, the framework loaded and evaluated lightweight IoT firmware in 210 s, including 170 s for static analysis. The usefulness was demonstrated when it was applied to 115 firmware images from 17 vendors and identified that 35 of them had memory opportunities for corruption, all of which were zero-day flaws.

Novak et al. (2019) [8] proposed an efficient partial firmware updating approach that has various advantages over existing methods. The quantity of data that must be transferred for an update is minimized, the energy efficiency is improved, and the radio spectrum utilization is lowered since the approach is designed for over-the-air updates. The solution presented here uses the Lua scripting interface to provide updatable native code pieces. This necessitates the use of a specific memory architecture, which is presented below. This approach may be used to deliver not just fixes for already installed systems, but also on-demand add-ons. Finally, the suggested firmware update system's security features are described, as well as its shortcomings.

The reliability of Microcontroller-based IoT firmware is investigated by Gao et al. (2019) [9]. Given the wide range of MCUs and their operating settings, the authors conducted case studies to find faults in current firmware upgrade models like PurpleAir's popular air quality sensor. On an ATmega1284P chip, the authors examined the concept of a secure firmware upgrading mechanism. To show the attack vector of the developed countermeasure, as well as the potential pitfalls found by this approach, as these problems may emerge during the deployment by other companies.

Jang et al. (2020) [10] investigated the behavior and vulnerabilities of Avatar and Firmadyne IoT firmware. While most analysis tools run firmware in a simulated space, there are several concerns to consider, ranging from whether firmware can be run to data gathering. The authors created a dynamic analytical model that can be used on IoT devices immediately. A dynamic analysis methodology based on S-Trace that operates on IoT architectures like MIPS and MIPSEL was presented, demonstrating that IoT vulnerabilities may be assessed effectively.

Wang et al. (2021) [11] presented a technique for scalable and smart firmware analysis. This technique examines over forty different firmware file system designs, summarizes, and retrieves file system characteristics, and creates a firmware feature repository. This solution can suit the demands of popular devices in terms of firmware analysis and security analysis. The suggested approach was tested on over 100 different types of genuine device firmware to ensure its usefulness. Passing our test, the enhanced program can now examine more than forty system files, substantially enhancing the firmware reverse investigation tool's capabilities. Zandberg (2019) [12] reported preliminary results using SUIT, a new IETF specification for secure IoT firmware upgrades. On several restricted commercial off-the-shelf IoT devices, we test the performance of our solution. The authors concluded that for IoT devices with less than 32 KB of RAM and 128 kB of flash memory, it is viable to provide a safe, standards-compliant automatic update system that employs cutting-edge security.

Tsai et al. (2020) [13] proposed a blockchain-based, real-time firmware updating architecture based on the MQTT protocol. As part of our framework's design, several MQTT servers are put in matched blockchain nodes to execute firmware patch delivery operations, allowing for automatic update operations on deployed IoT devices. In a blockchain network, smart contracts are used to find MQTT service providers and blockchain technology is utilized to keep track of who made a distributed firmware patch for a specific IoT device. Based on coding similarities, Wang et al. (2019) [14] provided a method for detecting staged firmware vulnerabilities. In the first phase, a neural network-based function embedding is employed to examine the analogies between functions, allowing for fast large-scale firmware security analysis. For fine-grained firmware security analysis, the second step calculates the similarity across function local call flow graphs, which can increase vulnerability identification accuracy. The experimental findings show that this approach is more efficient than state-of-the-art methods. Their method's average retraining time is 1 h, and the true positive rate of the top 30 in a real-world firmware vulnerability assessment experiment is as high as 86 percent.

Ahmed et al. (2020) [15] presented a scalable approach to verify the security of IoT firmware against the Mirai threat in this research paper. The authors put the proposed method to the test by statically analyzing over 1200 current firmware images to see how resistant they are to the Mirai botnet. According to the findings, the Mirai virus is present in 193 out of over 1200 firmware images. The authors tested this approach against a range of IoT device firmware images to get successful results. In comparison to other alternatives, the research determined that the approach is more extensible, less costly, and proactive.

Zheng et al. (2023) [16] proposed a lightweight incentive authentication scheme for forensic services in IoV. The objective was to increase IoT security and privacy while ensuring the convenience and motivation of data exchange among smart vehicles. This was developed on a three-tier architecture containing a cloud layer, fog layer, and user layer using pairing-free certificate-less sign encryption, pseudonym update mechanism, and incentive mechanism to realize a secure anonymous authentication efficiently. The authors conducted correctness

and security analysis, as well as performance analysis and evaluation to validate the high security and efficiency. Experimental results revealed that the communication and computation overheads as well as the message delay and packet loss are much lower than those of state-of-the-art techniques. The security and privacy concerns with forensic services in IoV for intelligent mobility served as the inspiration for this research.

With millions of gadgets being used every day in our homes and workplaces to improve our lives, the Internet of Things is constantly expanding. Through their sensor systems, smart devices become digital witnesses of our daily lives due to the strong coexistence between us and them. This opens a brand-new field of digital inquiry known as IoT Forensics, in which digital traces left behind by smart devices (mostly network traffic) are used as evidence in forensic investigations. To make the task of forensic investigators easier, it is crucial to develop technologies that can quickly acquire, preserve, and analyze such digital evidence. A tool called Feature-Sniffer developed specifically for Wi-Fi-enabled smart devices used in Smart Building/Smart Home scenarios, was presented by Palmese et al. (2022) [17]. For OpenWrt-based access points, Feature-Sniffer is an add-on that makes it simple to do online traffic feature extraction without having to store big PCAP files. With an accurate description of the technical details, Sagar et al. (2022) [18] introduced Feature-Sniffer and demonstrated its potential applications by using real-world examples for device identification and activity classification from encrypted traffic generated by IoT cameras. For reproducible research, the Feature-Sniffer was made available to the public.

The security concerns with Internet protocol cameras were the main topic of this study by Almazrouei et al. (2023) [19]. To help businesses and security specialists forecast attacker behavior and secure the systems, a deeper investigation of IP camera vulnerabilities and their impact on user security and privacy was conducted. The goal of this study was to investigate and identify the security and privacy flaws related to an IP camera. To do this, a direct inspection of the camera was conducted. A VAVA Outdoor Wireless IP Security Cam that was used as a home security camera was put to the test in a real-world setting. For this, data was acquired from a variety of online sources, and after that, the hardware and software were used to check the device's security characteristics. The study's conclusions showed that the IP camera has security flaws and openings that jeopardize user security.

By examining the characteristics of unidirectional packets contained in non-TLS network traffic, Liu et al. (2023) [20] developed a novel method to address IoT camera security issues. The scientists discovered that manually classifying traffic data based on device behavior takes effort and is frequently erroneous. To solve this issue, this research developed a two-factor automatic behavior tagging approach based on reverse traffic and operation logs. The authors used a real-world dataset and the public IMC 2019 payload dataset to test the performance of various classifiers. The suggested system can automatically extract and detect behavioral traffic, according to experiments. Based on aspects of unidirectional upload traffic, the CNN-based classifier can precisely identify HSC devices' fine-grained behavioral activity.Fig. 1.,Fig. 2. Fig. 3..

3. Research methodology

Smart IoT devices used in homes and industries, interact with firmware to deliver device functions and features. The firmware can be looked at as the actual source code inside the device OS running the IoT device. This security analysis involved twelve unique steps from selecting the IoT camera device and acquiring the firmware image to analyzing the filesystem using various flags, command options flags, and unique tools to finally search for sensitive and interesting information. Fig. 4 below illustrates a unique step-by-step approach as the proposed research methodology for this research.

Below is a list of some popular IoT camera firmware for smart homes reviewed for this research:

1. Wyze: Wyze is an IoT camera firmware that offers a range of smart home products, including security cameras, smart plugs, and smart sensors. It is compatible with Amazon Alexa and Google Assistant

Fig. 2. D-Link DSC-5020L IoT Camera [16].

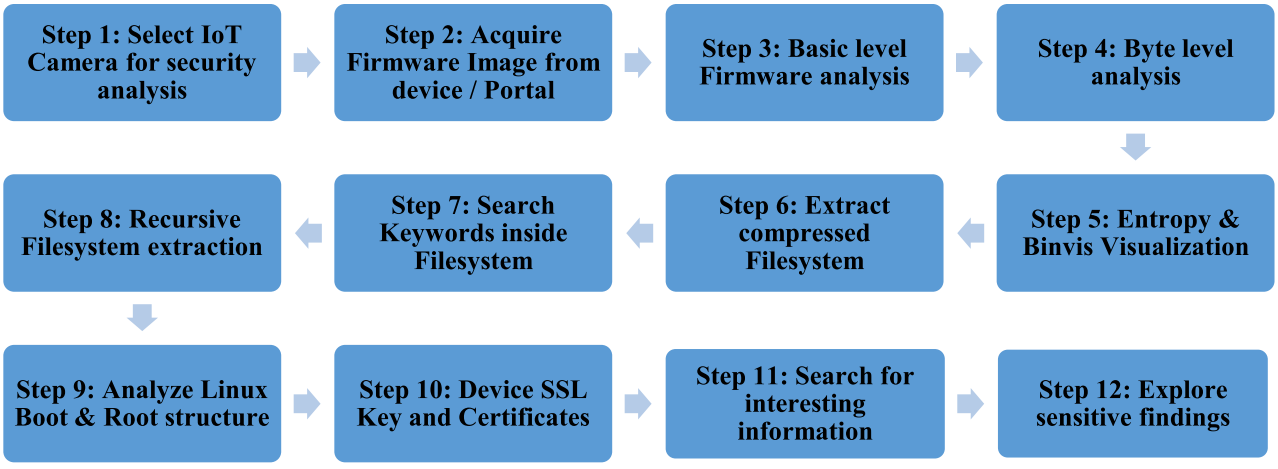

Fig. 1. Comprehensive IoT Firmware Security Analysis.

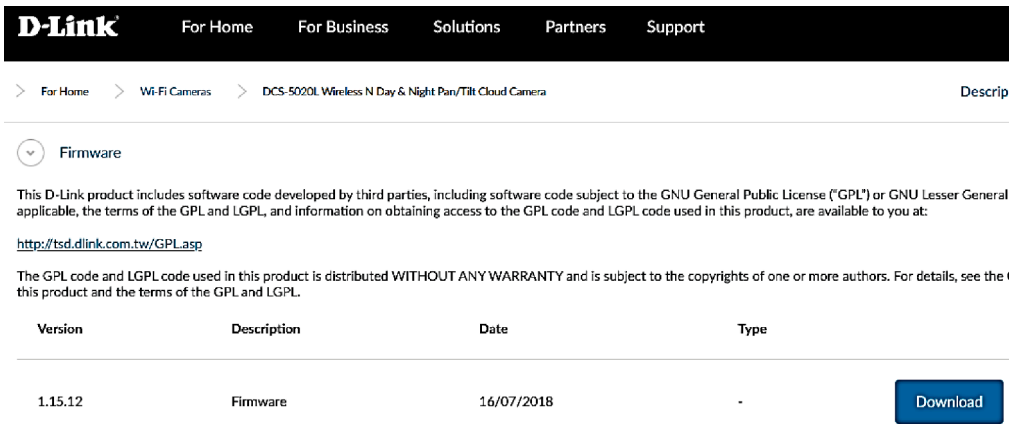

Fig. 3. D-Link Support portal to download firmware.

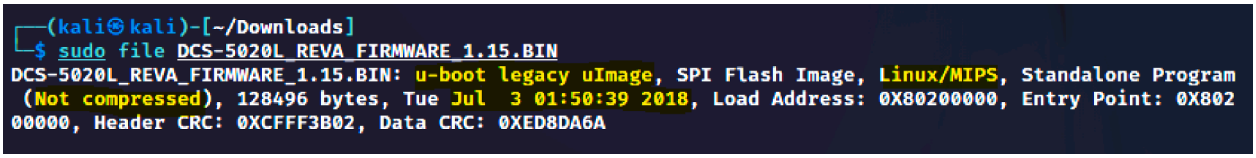

Fig. 4. Initial level of analysis.

and allows users to monitor and control their smart home devices remotely.

2. Netatmo: Netatmo is an IoT camera firmware that offers a range of smart home products, including security cameras, smart thermostats, and smart smoke detectors. It is compatible with Amazon Alexa and Google Assistant and allows users to monitor and control their smart home devices through a smartphone app.
3. Arlo: Arlo is an IoT camera firmware that offers a range of smart home products, including security cameras, smart doorbells, and smart locks. It is compatible with Amazon Alexa and Google Assistant and allows users to monitor and control their smart home devices through a smartphone app.
4. Blink: Blink is an IoT camera firmware that offers a range of smart home products, including security cameras, smart doorbells, and smart locks. It is compatible with Amazon Alexa and Google Assistant and allows users to monitor and control their smart home devices through a smartphone app.
5. Nest: Nest is an IoT camera firmware that offers a range of smart home products, including security cameras, smart thermostats, and smart smoke detectors. It is compatible with Amazon Alexa and Google Assistant and allows users to monitor and control their smart home devices through a smartphone app.

It is important to note that these are just a few examples of IoT camera firmware for smart homes, and there are many other options available on the market. It is important to research and compare different options to find the best fit for your needs. Several features can be referred to for comparison, namely price range, ease of installation, video resolution, field of view, night vision, motion detection, two-way audio, Cloud storage, local storage, smart-home, and mobile app integration. Table 1 presents the comparative analysis of IoT camera firmware for smart homes concerning the relevant features for this research.

4. Security analysis performed

4.1. Select IoT camera for security analysis

The first step of the security analysis is selecting an IoT device. The authors decided to experiment and analyze commonly found Smart IoT camera devices in homes. One such device is the fully-featured surveillance smart IoT D-Link DCS-5020L wireless network camera [21]. This IoT device works day & night and offers several features like wide view range, 4x digital zoom, motion detection, pan and 120-degree tilt, mobile app, and web browser access for remote access. The device also has a built-in wireless extender to expand Wi-Fi coverage up to 8 m.

Table 1
Comparative Analysis of IoT cameras firmware for smart homes.

Feature	Wyze	Netatmo	Arlo	Blink	Nest
Compatibility	Alexa, Google Assistant	Alexa, Google Assistant	Alexa, Google Assistant	Alexa, Google Assistant	Alexa, Google Assistant
Smart Home Devices	Security cameras, smart plugs, smart sensors	Security cameras, smart thermostats, smart smoke detectors	Security cameras, smart doorbells, smart locks	Security cameras, smart doorbells, smart locks	Security cameras, smart thermostats, smart smoke detectors
Remote Monitoring and Control	Yes	Yes	Yes	Yes	Yes
Mobile App	Yes	Yes	Yes	Yes	Yes
Price	$	$	$	$	$

Note: The price column is based on a rough estimation, with $ representing a low price, $$ representing a moderate price, and $$$ representing a high price. The actual price may vary depending on the specific product and features. This table compares the compatibility, range of smart home devices, and remote monitoring and control capabilities of different IoT camera firmware for smart homes. It also includes a rough estimation of the price range for each firmware. This table can be used as a reference for comparing the features of different IoT camera firmware and determining which one might be the best fit.

4.2. Acquire firmware image from device or portal

To acquire the firmware image file for this IoT camera, the authors downloaded the firmware from the D-Link Support portal [22]. The firmware can be downloaded and distributed without a warranty and includes code developed by third parties.

4.3. Basic-level firmware analysis

Linux provides a rich set of native tools to analyze and learn about the downloaded file, in this case, the D-Link camera firmware. Features like whether the file is compressed, encrypted, or corrupted and if it is a binary executable, ASCII, Video, or Image file. This basic analysis starts with the use of the Linux 'file' command, which determines the uncompressed file type, on MIPS architecture having u-boot legacy image created on 3rd June 2018.

4.4. Byte level analysis

To validate the above findings and gather some basic information, the Binwalk tool [23] provides offset content details about the image file. Fig. 5 confirms this firmware file is a Linux OS running MIPS architecture with the 'U-Boot' bootloader at the 99,360 offset another uImage offset at 327,744 and the filesystem in LZMA compressed format [24].

5. Results obtained

Section 4 performed the basic level analysis of the firmware, and section 5 highlights the deep-level assessment and security analysis searching for interesting and sensitive information which can be used to exploit the Smart Home IoT devices in the future.

5.1. Entropy and Binvis Visualization

Entropy analysis helps determine whether the entire file system is encrypted or compressed. This measures the randomness concerning areas that acquire random data, which are the bits required for representing each character in the file. The higher the entropy, the higher the probability of the file being encrypted or compressed. This reveals the amount of information present in the file. Entropy is a measure of the randomness or uncertainty in a system. In the context of IoT firmware, entropy could refer to the randomness or unpredictability of the data being transmitted, the complexity of the firmware code, or the degree of randomness in the inputs to the firmware. To mathematically define entropy, we can use the concept of information entropy, which is a measure of the amount of uncertainty or randomness in a message or system. Information entropy is often represented by the symbol H and is typically defined as the negative sum of the probabilities of all possible outcomes multiplied by the algorithm of those probabilities:

$$H = -\sum p \tag{1}$$

This equation is known as Shannon entropy, often denoted as H, and is a fundamental concept in information theory. It is used to measure the uncertainty or average amount of information contained in a probability distribution where:

H: Shannon entropy, which represents the amount of uncertainty or information content in a probability distribution.

p(x): The probability of an event × occurring. In the context of this equation, x represents a specific outcome or value from a set of possible outcomes.

$\sum$: Sigma notation denotes summation, which means you need to sum up the values of the expression that follows it.

For example, consider a simple IoT device that transmits a message containing a binary digit (0 or 1). The entropy of this system would be:

$$H = -(0.5*log(0.5)) - (0.5*log(0.5)) = 1\,bit \tag{2}$$

This represents the maximum amount of uncertainty or randomness that can be transmitted in a single message. If the IoT device transmitted a message containing two binary digits (00, 01, 10, or 11), the entropy would be:

$$H = -(0.25*log(0.25)) - (0.25*log(0.25)) - (0.25*log(0.25)) - (0.25*log(0.25)) = 2\,bits \tag{3}$$

These equations represent the maximum amount of uncertainty or randomness that can be transmitted in a message containing two binary digits. In general, the entropy of an IoT firmware system can be quantified by considering the complexity and randomness of the data being transmitted, the complexity of the firmware code, and the degree of randomness in the inputs to the firmware. Executing the 'Binwalk' tool with the '-E' flag option illustrates the Entropy output as presented in Fig. 6 to plot. This reveals the firmware file having high entropy,

```
┌──(kali㉿kali)-[~/Downloads]
└─$ sudo binwalk DCS-5020L_REVA_FIRMWARE_1.15.BIN

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             uImage header, header size: 64 bytes, header CRC: 0xCFFF3B02, created: 20
18-07-03 01:50:39, image size: 128496 bytes, Data Address: 0x80200000, Entry Point: 0x80200000, data CR
C: 0xED8DA6A, OS: Linux, CPU: MIPS, image type: Standalone Program, compression type: none, image name:
 "SPI Flash Image"
99360         0x18420         U-Boot version string, "U-Boot 1.1.3"
100000        0x186A0         CRC32 polynomial table, little endian
115680        0x1C3E0         HTML document header
116026        0x1C53A         HTML document footer
116036        0x1C544         HTML document header
116228        0x1C604         HTML document footer
116396        0x1C6AC         HTML document header
117089        0x1C961         HTML document footer
327680        0x50000         uImage header, header size: 64 bytes, header CRC: 0xBC0D737A, created: 20
18-07-03 01:50:32, image size: 7301445 bytes, Data Address: 0x80000000, Entry Point: 0x80391000, data C
RC: 0x6FAA218C, OS: Linux, CPU: MIPS, image type: OS Kernel Image, compression type: lzma, image name:
"Linux Kernel Image"
327744        0x50040         LZMA compressed data, properties: 0x5D, dictionary size: 33554432 bytes,
uncompressed size: 9919699 bytes
3958733       0x3C67CD        MySQL ISAM compressed data file Version 5
```

Fig. 5. Basic Binwalk analysis.

indicating a compressed file system, not encrypted. The file entropy is majorly close to 1 (high) with dips during the initial and end.

Besides Entropy, the authors also performed full-color binary visualization as illustrated in Fig. 7 using Binvis [25].

5.2. Extract compressed filesystem

Since the firmware is compressed, 'dd' command [26] converts and copies files, and the requisite standard input is sent into a standard output file. The output includes the in-out size records and counts bytes along with the skip field from the decimal offset determined from the Binwalk output as presented in Fig. 8.

5.3. Search keywords inside filesystem

Now the decompressed and extracted firmware is searched for interesting information using the 'strings' command utility [27], filtering for specific keywords such as 'squashfs' or 'CIFS'. Fig. 9 confirms the D-Link IoT camera is using the SquashFS file system. This is a highly compressed, Linux-based read-on filesystem, the advantage is more flexibility and performance as compared to GZ or TAR formats. So, this file holds the Linux root file within another live OS folder having the rootfs.

5.4. Recursive filesystem extraction

Recursive extraction of the root filesystem including files inside subfolders of the firmware file is performed using the '-eM' flag in the Binwalk tool. This extracts the files that were identified during the initial file signature scan as illustrated in Fig. 10. The initial scan reports 411 signatures and confirms MIPS CPU architecture, no encryption, LZMA compression, and Linux operating system.

On checking the ''-eM' scan output, Fig. 11 illustrates the copyright string along with paths for the /var, /etc, and /usr folders and filenames for further investigations.

5.5. Analyze Linux boot and root structure

Traversing the folders one by one, the root directory of the filesystem is found to be in the 'cpio-root' as presented in Fig. 12.

5.6. Device SSL key and certificates

The '/etc_ro' folder is found to have the camera's SSL key, the code contents of which can be viewed using 'grep' and 'cat' commands along with some telnet binaries are found to be installed as presented in Fig. 13 and a new certificate can be generated as the HTTPS is enabled on the

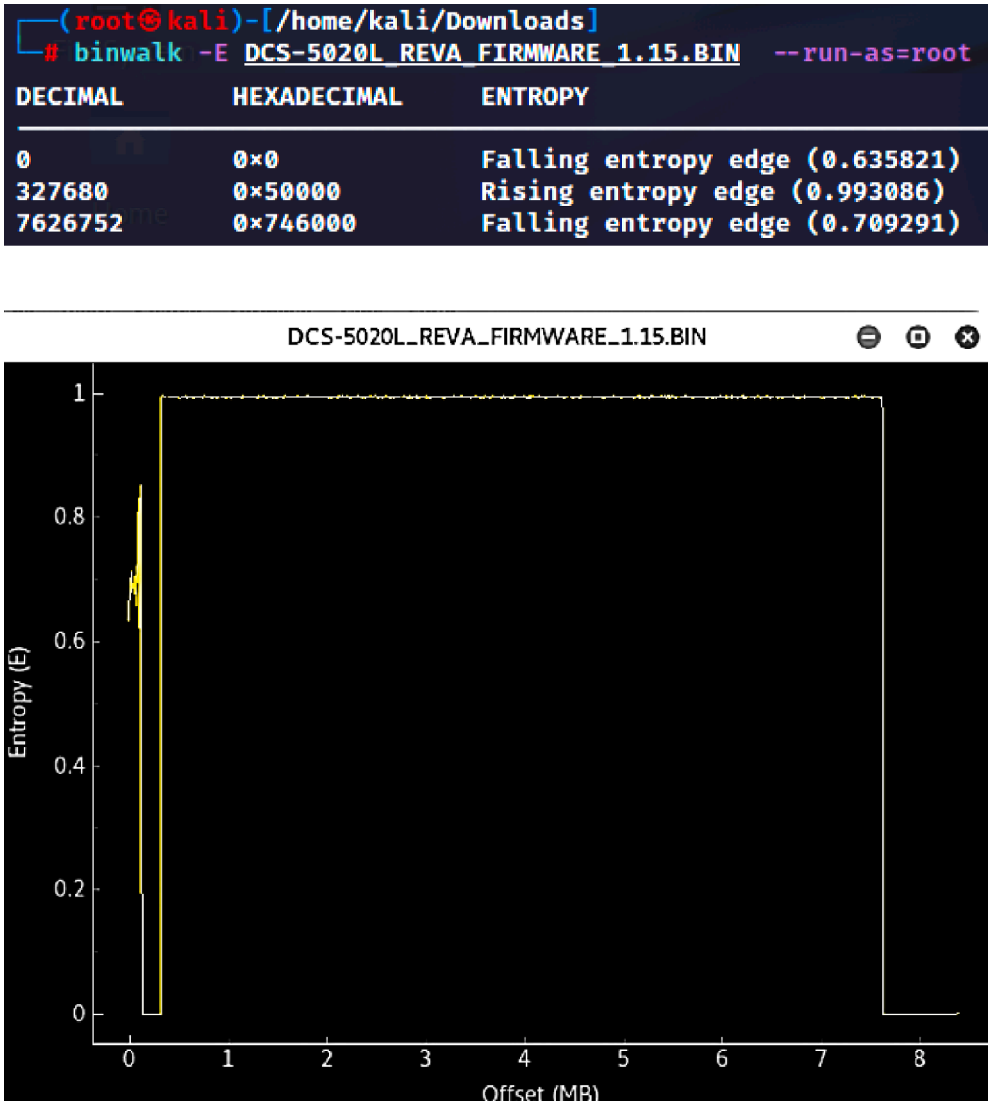

Fig. 6. Entropy Output and Plot Visualization for rising & falling edge.

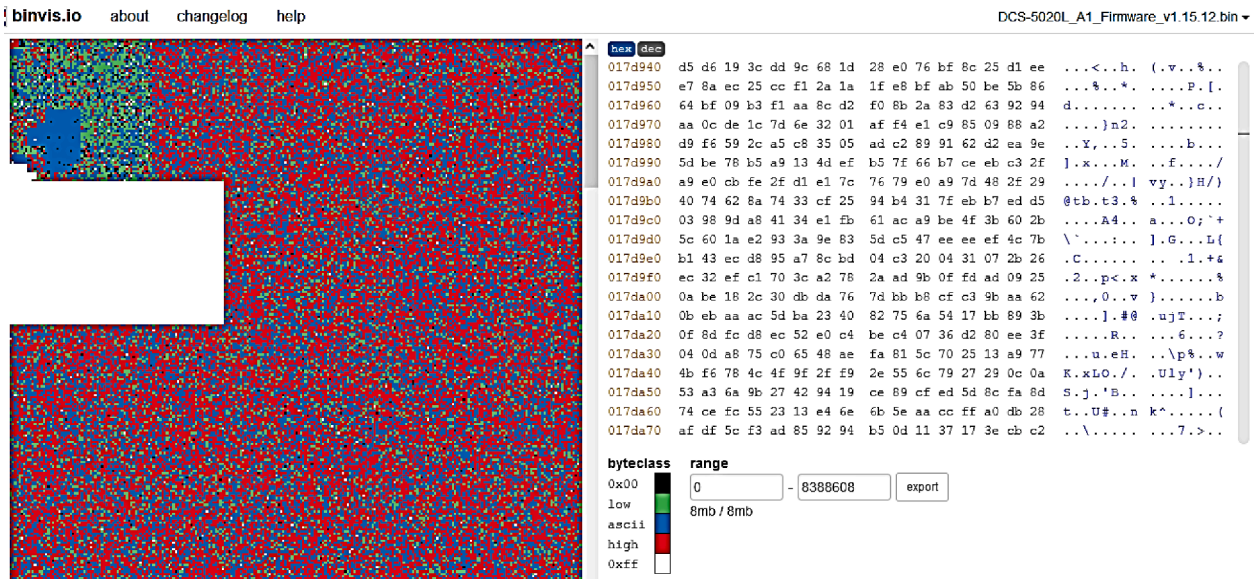

Fig. 7. Binary Visualization.

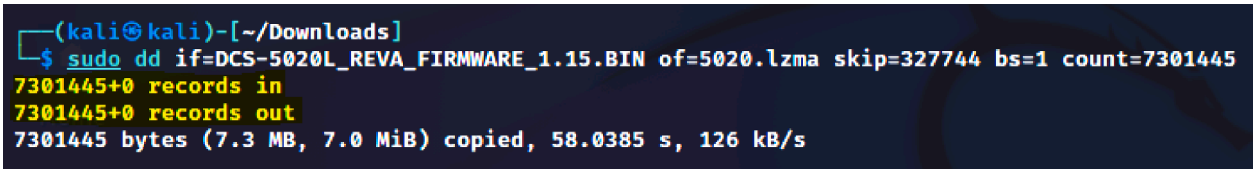

Fig. 8. Use of the 'dd' command for byte-level analysis.

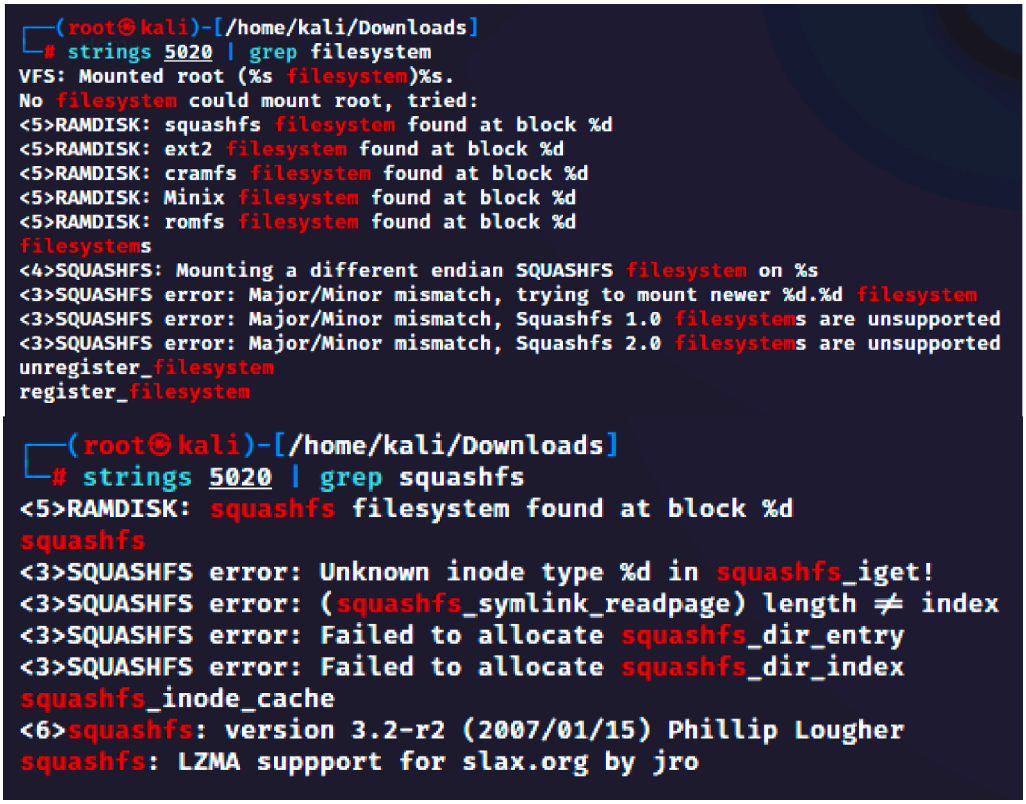

Fig. 9. Validation of IoT device filesystem.

```
┌──(root㉿kali)-[/home/kali/Downloads]
└─# binwalk -eM DCS-5020L_REVA_FIRMWARE_1.15.BIN --run-as=root

Scan Time:     2022-04-04 09:00:05
Target File:   /home/kali/Downloads/DCS-5020L_REVA_FIRMWARE_1.15.BIN
MD5 Checksum:  5a5e467bf4110f9fa8175f50aba4818b
Signatures:    411

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             uImage header, header size: 64 bytes, header CRC: 0xCFFF3B02, created: 2018-07-03
01:50:39, image size: 128496 bytes, Data Address: 0x80200000, Entry Point: 0x80200000, data CRC: 0xED8DA6A, OS:
Linux, CPU: MIPS, image type: Standalone Program, compression type: none, image name: "SPI Flash Image"
99360         0x18420         U-Boot version string, "U-Boot 1.1.3"
100000        0x186A0         CRC32 polynomial table, little endian
115680        0x1C3E0         HTML document header
116026        0x1C53A         HTML document footer
116036        0x1C544         HTML document header
116228        0x1C604         HTML document footer
116396        0x1C6AC         HTML document header
117089        0x1C961         HTML document footer
327680        0x50000         uImage header, header size: 64 bytes, header CRC: 0xBC0D737A, created: 2018-07-03
01:50:32, image size: 7301445 bytes, Data Address: 0x80000000, Entry Point: 0x80391000, data CRC: 0x6FAA218C, OS
: Linux, CPU: MIPS, image type: OS Kernel Image, compression type: lzma, image name: "Linux Kernel Image"
327744        0x50040         LZMA compressed data, properties: 0x5D, dictionary size: 33554432 bytes, uncompres
sed size: 9919699 bytes
3958733       0x3C67CD        MySQL ISAM compressed data file Version 5
```

Fig. 10. Firmware Extraction using –eM flag.

```
DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
3276876       0x32004C        Linux kernel version 2.6.21
3317264       0x329E10        SHA256 hash constants, little endian
3323216       0x32B550        AES Inverse S-Box
3323984       0x32B850        AES S-Box
3364384       0x335620        Unix path: /usr/gnemul/irix/
3366308       0x335DA4        Unix path: /usr/lib/libc.so.1
3423052       0x343B4C        Copyright string: "Copyright (c) 2011 Alpha Networks Inc."
3432948       0x3461F4        Unix path: /var/run/udhcpc.pid
3508480       0x358900        Unix path: /usr/bin/killall
3511744       0x3595C0        Unix path: /etc/Wireless/RT2860STA/e2p.bin
3520464       0x35B7D0        Unix path: /etc/Wireless/RT2860STA/RT2860STA.dat
3520596       0x35B854        Unix path: /etc/Wireless/RT2860/RT2860.dat
3570923       0x367CEB        Neighborly text, "neighbor %.2x%.2x.%.2x:%.2x:%.2x:%.2x:%.2x:%.2x lost on port %d(
%s)(%s)"
3679248       0x382410        CRC32 polynomial table, little endian
3682752       0x3831C0        AES S-Box
3874816       0x3B2000        LZMA compressed data, properties: 0x5D, dictionary size: 1048576 bytes, uncompress
ed size: 16473088 bytes
```

Fig. 11. Recursive Files and Folders extracted.

```
┌──(root㉿kali)-[/home/…/Downloads/_5020.extracted/_3B2000.extracted/cpio-root]
└─# ls -l
total 60
drwxr-xr-x  2 501 501 4096 Apr  4 09:38 bin
drwxrwxr-x  3 501 501 4096 Apr  4 09:38 dev
drwxrwxr-x  2 501 501 4096 Apr  4 09:38 etc
drwxrwxr-x 10 501 501 4096 Apr  4 09:38 etc_ro
drwxrwxr-x  3 501 501 4096 Apr  4 09:38 home
lrwxrwxrwx  1 501 501   11 Apr  4 09:38 init → bin/busybox
drwxr-xr-x  4 501 501 4096 Apr  4 09:38 lib
drwxrwxr-x  2 501 501 4096 Apr  4 09:38 media
drwxrwxr-x  2 501 501 4096 Apr  4 09:38 mnt
drwxrwxr-x  2 501 501 4096 Apr  4 09:38 mydlink
drwxrwxr-x  2 501 501 4096 Apr  4 09:38 proc
drwxrwxr-x  2 501 501 4096 Apr  4 09:38 sbin
drwxrwxr-x  2 501 501 4096 Apr  4 09:38 sys
drwxrwxr-x  2 501 501 4096 Apr  4 09:38 tmp
drwxrwxr-x  5 501 501 4096 Apr  4 09:38 usr
drwxrwxr-x  2 501 501 4096 Apr  4 09:38 var
```

Fig. 12. Extracted Root directory Filesystem.

config page. This can be used to take over and control the IoT device as part of advanced-level attacks.

5.7. Search for interesting information

With the root filesystem and folders extracted, the Firmwalker tool helps analyze the firmware by searching for sensitive contents like usernames, passwords, emails, private keys, and IP addresses as illustrated in Fig. 14.

5.8. Explore sensitive findings

From the Firmwalker output, the authors performed a direct filesystem lookup. Fig. 15 illustrates the SSL-related certificate files including the server certificate inside the cpoi-root/etc or folder of the Linux kernel. The authors executed and validated the shell scripts.

A few folders containing details about 'admin', 'root and 'password', 'passwd' were extracted. These files are of high critical risk as these contain hard-coded usernames and passwords, all residing inside the extracted firmware as illustrated in Fig. 16.

Most IoT devices run the web user interface on different ports, including HTTP (port 80), SSL (port 443), and Telnet (port 23). The Binwalk output validates the same in Fig. 17 as does Shodan (IoT Search engine) displaying at least 1007 live D-Link IoT Cameras running 5050L model on the Internet.

The authors even found IP Addresses and URLs inside the firmware as illustrated in Fig. 18. This amounts to huge critical information disclosure as such information can be useful for conducting advanced-level device take-over attacks.

The authors filtered and analyzed the URL domains using VirusTotal API with a simple bash script as presented in Fig. 19 for domain categories, and methods, if they are malicious or harmless. Fig. 20 presents the hardcoded email addresses that are embedded in the filesystem.

Binwalk also uncovered a set of hardcoded email addresses, which pose a significant threat to operational continuity because they are rarely altered and can be simply spoofed to compromise IoT devices. In 2016, the Mirai virus scanned Telnet services on Linux-based IoT devices [28]. Hackers then attempted to get in using a table of 61 predefined default user credentials using a brute force attack. Mirai and its variants were used to create enormous botnets of IoT systems, with up to 400,000 connected devices, the majority of which had no idea who they belonged to. Mirai-related botnets wreaked havoc on French Telecom, Krebs on

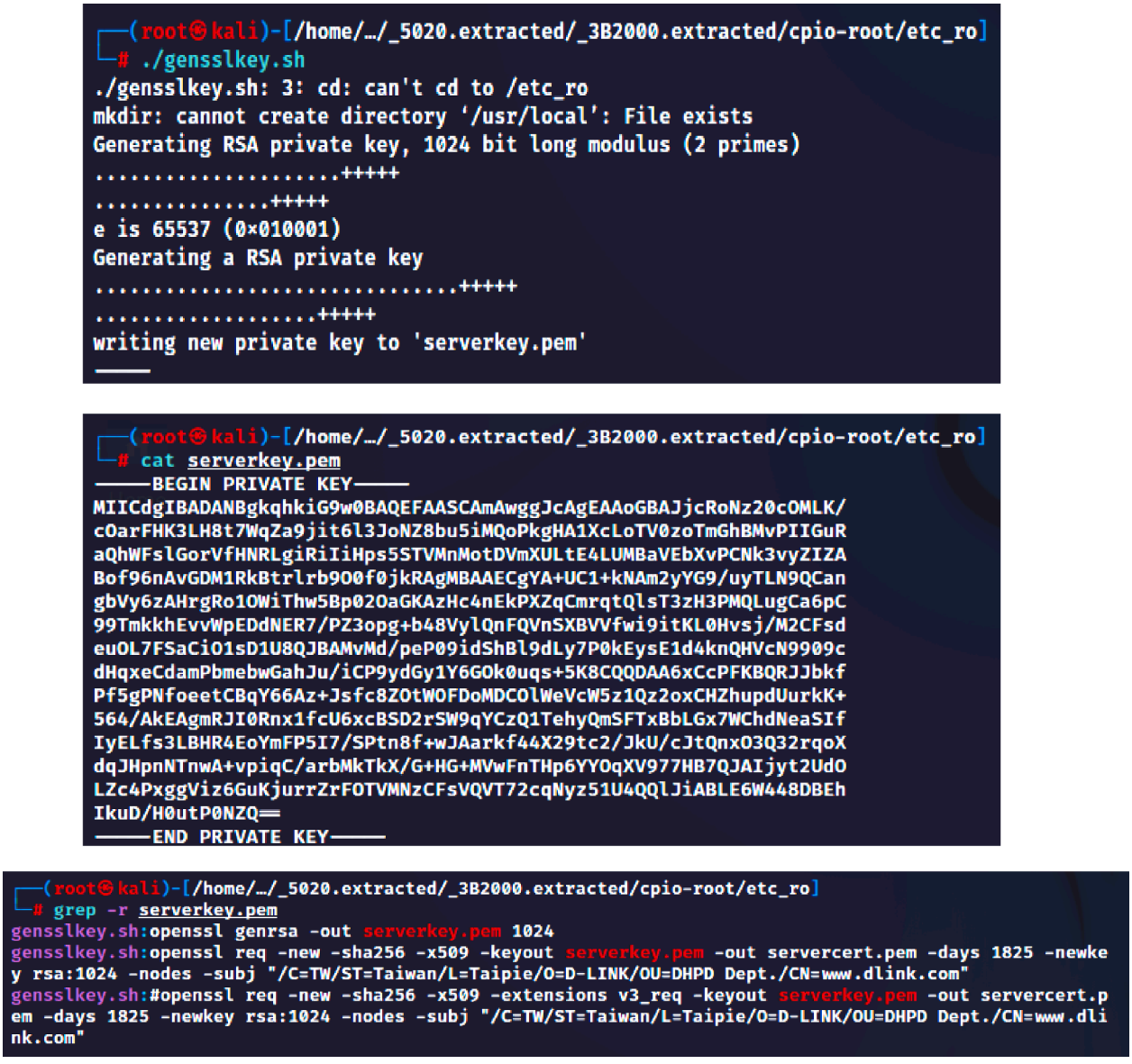

Fig. 13. Private Key found & generated new Certificate.

```
┌──(root㉿kali)-[/home/kali/Downloads/_DCS-5020L_REVA_FIRMWARE_1.15.BIN.extracted/firmwalker]
└─# shodan init ZFrdUeutbsobce4AQCYlojjAFDDJTjWl
Successfully initialized

┌──(root㉿kali)-[/home/kali/Downloads/_DCS-5020L_REVA_FIRMWARE_1.15.BIN.extracted/firmwalker]
└─# ./firmwalker.sh /home/kali/Downloads/_5020.extracted
**Firmware Directory**
/home/kali/Downloads/_5020.extracted
```

Fig. 14. Firmwalker search for sensitive information.

```
**Search for SSL related files**
################################### *.crt
d/_3B2000.extracted/cpio-root/mydlink/pub.crt

################################### *.pem
d/_3B2000.extracted/cpio-root/etc_ro/servercert.pem
d/_3B2000.extracted/cpio-root/etc_ro/serverkey.pem

**Search for shell scripts**
################################### shell scripts
d/_3B2000.extracted/cpio-root/mydlink/mydlink-watch-dog.sh
d/_3B2000.extracted/cpio-root/etc_ro/gensslkey.sh
d/_3B2000.extracted/cpio-root/sbin/ppp-loop.sh
d/_3B2000.extracted/cpio-root/sbin/nat.sh
d/_3B2000.extracted/cpio-root/sbin/video.sh
d/_3B2000.extracted/cpio-root/sbin/hso_connect.sh
d/_3B2000.extracted/cpio-root/sbin/vpn-passthru.sh
d/_3B2000.extracted/cpio-root/sbin/lan.sh
d/_3B2000.extracted/cpio-root/sbin/snort.sh
```

Fig. 15. SSL-related certificate files and shell scripts.

Security, Dyn, Deutsche Telecom, Russian banks, and the Liberian government, among others.

6. Societal context of the research

The significance of this research extends far beyond the realm of technological analysis. Its implications resonate profoundly in the societal context, as it addresses critical issues related to privacy, security, and the broader impact of IoT devices on our daily lives. The proliferation of IoT devices, especially in smart homes, has brought convenience and connectivity to an unprecedented level. IoT cameras are at the forefront of this revolution, promising enhanced security, remote monitoring, and automation. However, this convenience comes with a caveat - the potential compromise of privacy and security. This research endeavors to shed light on these implications and their societal consequences.

Privacy is a fundamental human right, and the advent of IoT cameras in homes has raised pertinent questions about the boundaries between personal and public spaces. These devices, if compromised, can intrude into the most intimate aspects of individuals' lives. The significance of this research lies in its quest to uncover vulnerabilities in IoT camera firmware, vulnerabilities that, if exploited, could lead to breaches of privacy. By identifying and addressing these issues, society can better safeguard its right to privacy in an increasingly connected world. Moreover, the research contributes significantly to enhancing cybersecurity awareness and preparedness. As IoT cameras become more integrated into our homes, they become attractive targets for cybercriminals. Vulnerabilities in firmware can lead to unauthorized access, data breaches, and even the use of these devices in large-scale cyberattacks. By conducting a forensic analysis and security assessment of IoT camera firmware, this research not only helps protect individual homes but also contributes to the collective cybersecurity resilience of society.

In a broader societal context, this research underscores the importance of consumer awareness and informed decision-making. As consumers, we are often drawn to the latest technological innovations without fully understanding their implications. This research empowers individuals to make informed choices about the IoT devices they bring into their homes, promoting responsible consumption and ensuring that technology serves rather than subverts societal interests. Furthermore, this research has implications for the regulatory landscape. Governments and regulatory bodies must keep pace with technological advancements to protect their citizens adequately. By exposing vulnerabilities and shortcomings in IoT camera firmware, this research can inform the development of robust regulations and standards that ensure the security and privacy of consumers.

Additionally, the societal context includes issues of trust and accountability. Manufacturers and service providers play a pivotal role in the deployment of IoT devices. Through this research, the accountability of these entities in ensuring the security and privacy of their products is highlighted. It encourages responsible practices in the IoT industry and fosters trust between consumers and technology providers. Lastly, the significance of this research extends to its role in shaping future technological developments. Identifying weaknesses in current

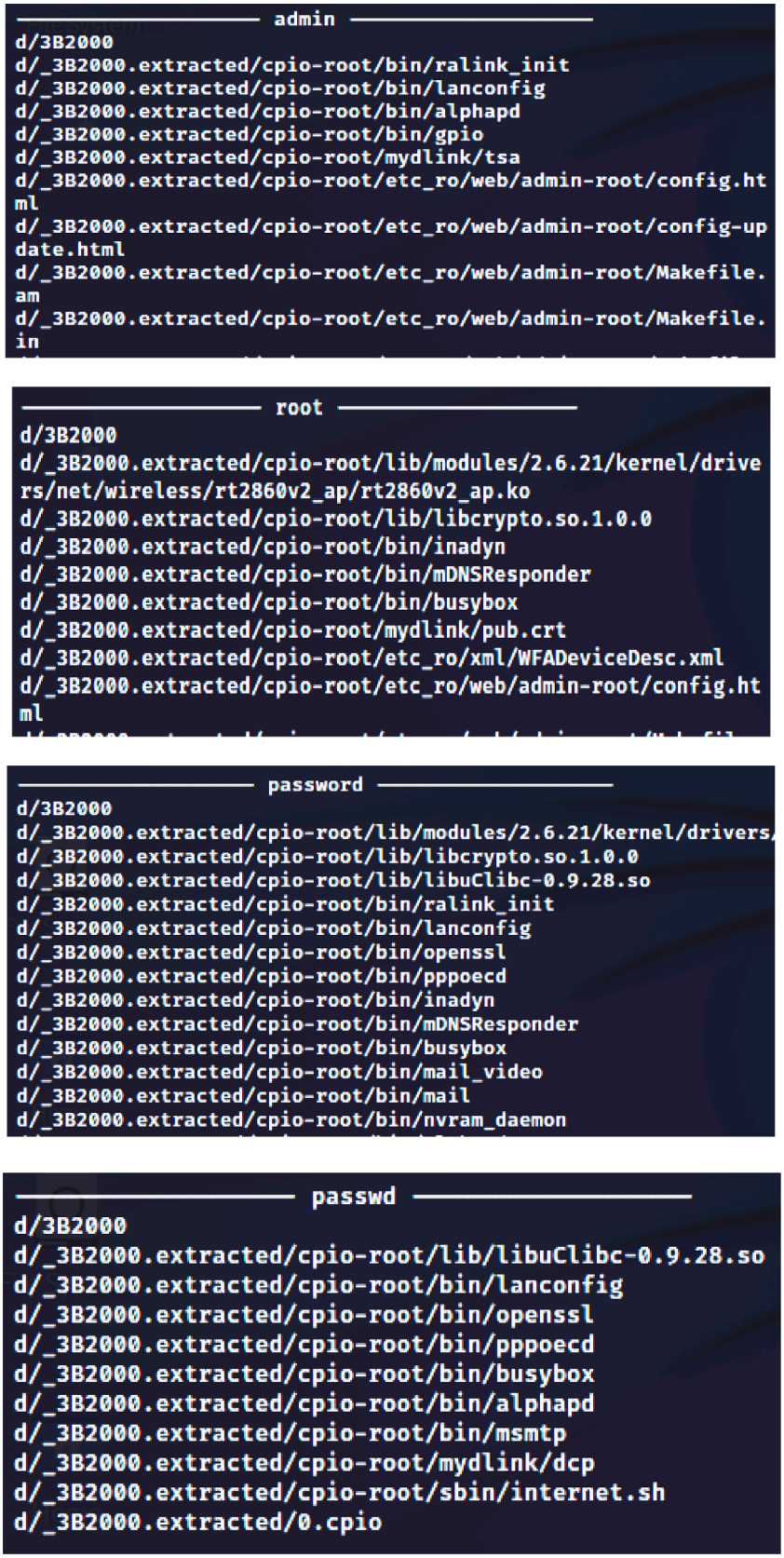

Fig. 16. Password & Passwd files.

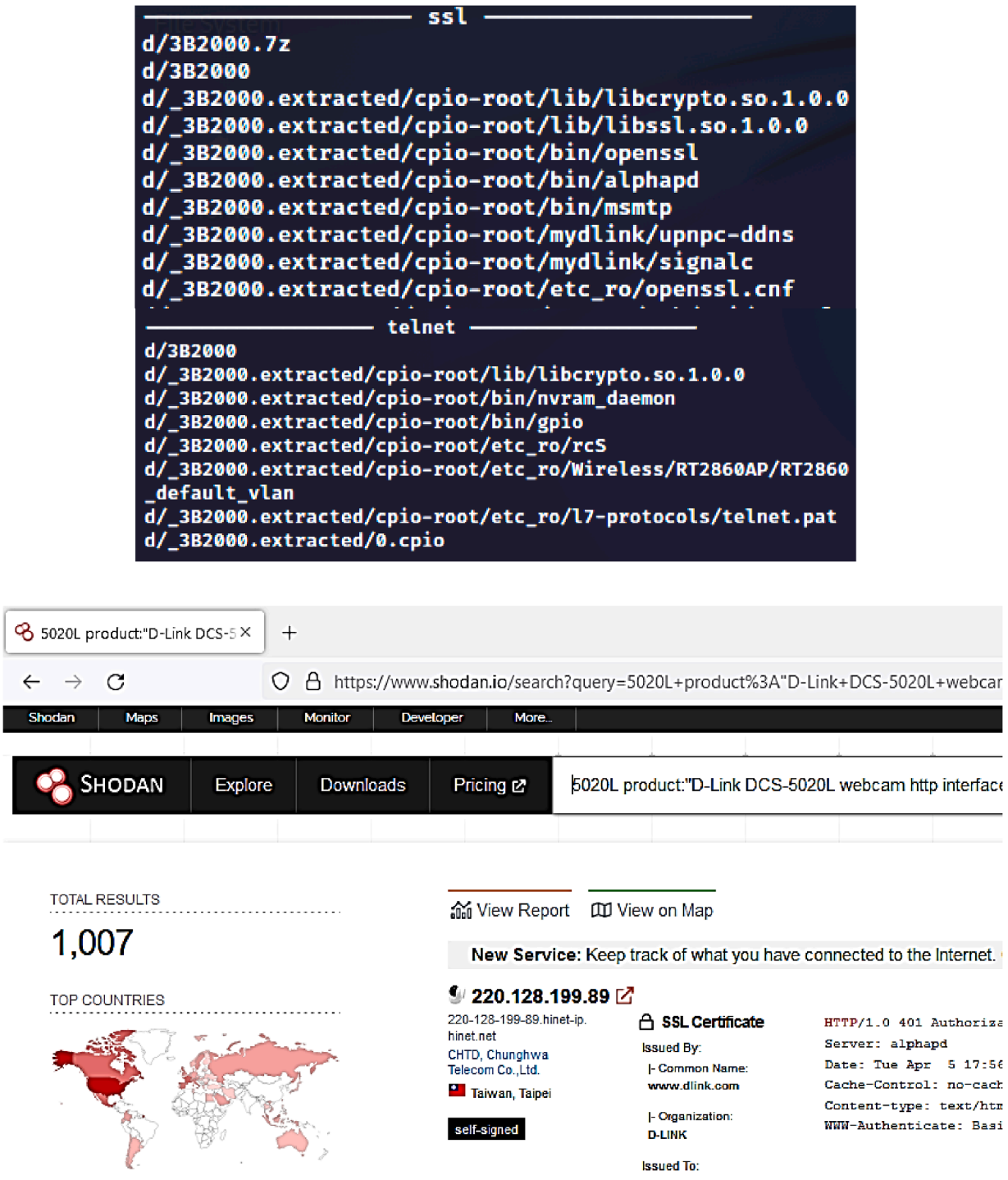

Fig. 17. SSL & Telnet files in Firmware and Live devices on Shodan.

IoT camera firmware serves as a catalyst for innovation and improvement in the industry. As technology evolves, lessons learned from this research can inform the development of more secure and privacy conscious IoT devices, ultimately benefiting society at large.

When security vulnerabilities exist at the firmware level, they are difficult to detect and locate, and their effect quickly spreads across many devices. In the automotive, healthcare, manufacturing, and consumer industries, manufacturers, suppliers, and system integrators of connected devices should base their firmware implementation analysis on appropriate standards and regulations for their products. While the Mirai assaults were important in terms of disrupting business, the Uber data breach [29] compromised the personally identifiable information of 57 million customers and 600,000 drivers. Hardcoded credentials, just like with Mira, were to blame. Unencrypted credentials were provided in source code by an Uber employee, which was then uploaded on Github, a popular developer source. A competent hacker found the encoded credentials on GitHub and used them to obtain privileged access to Uber's Amazon AWS Instances.

In conclusion, this research holds immense significance in the societal context. It safeguards privacy, enhances cybersecurity, empowers consumers, informs regulation, promotes accountability, and drives technological progress. In an era defined by rapid technological advancement, this research serves as a beacon, guiding us toward a future where the benefits of IoT devices are harnessed without

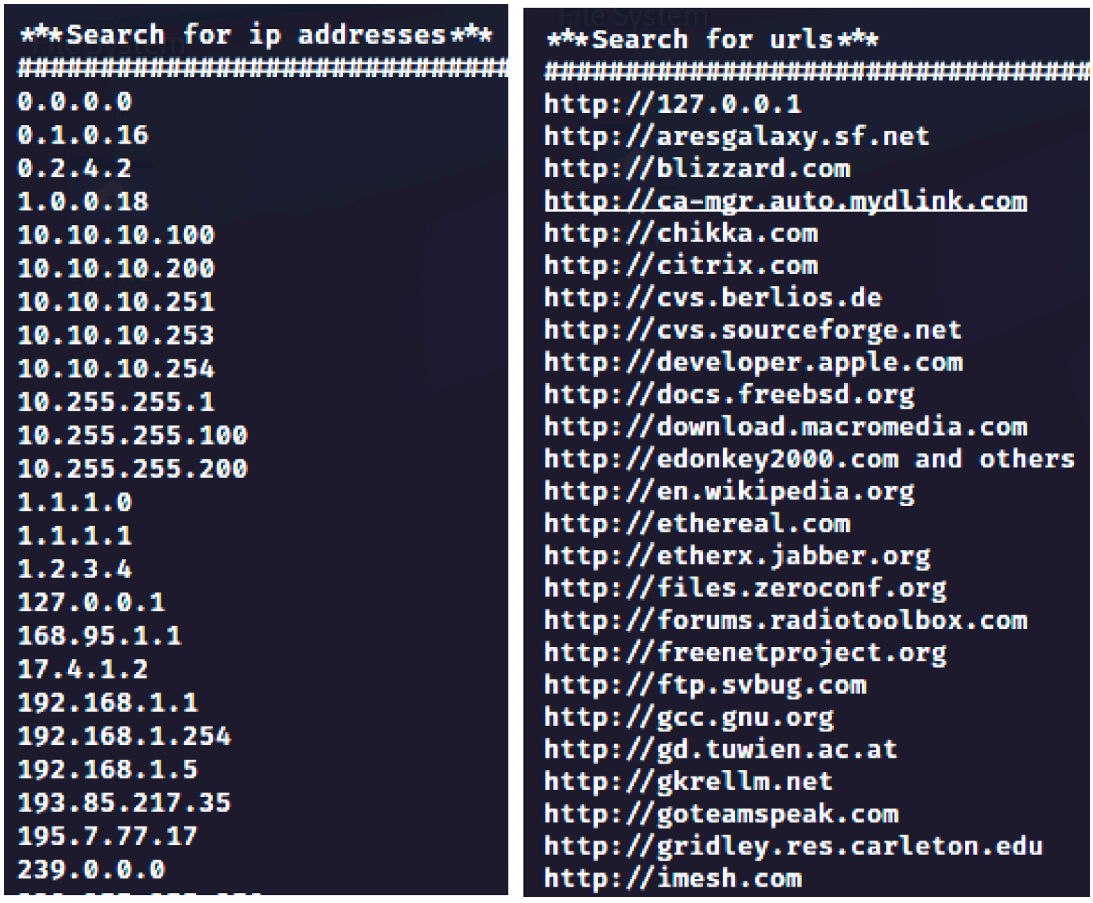

Fig. 18. IP Addresses and Domains information.

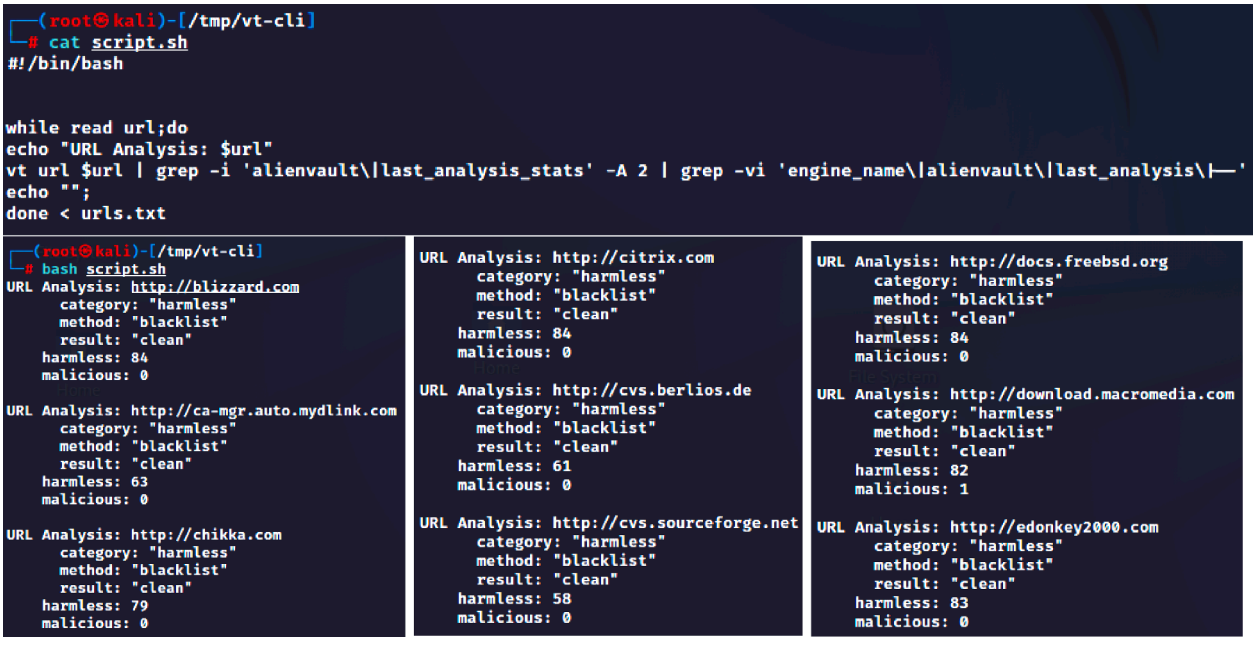

Fig. 19. Script output for URL domain results.

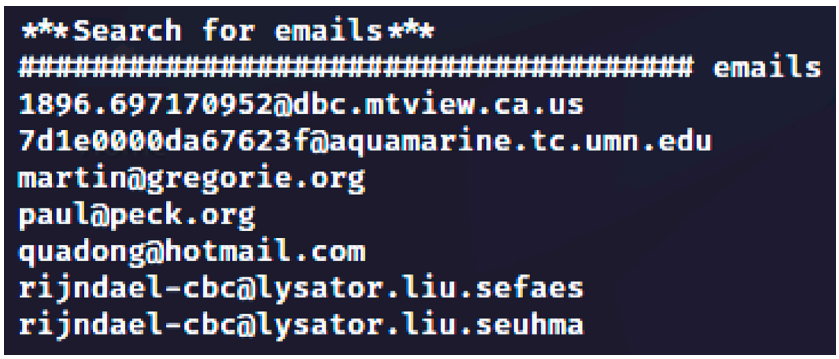

Fig. 20. Email addresses found in Filesystem.

compromising our fundamental rights and societal well-being.

7. Conclusion

Though IoT deployments are increasing across a variety of sectors, most IoT devices do not have a built-in secure firmware upgrading mechanism. Critical security flaws cannot be addressed without such a mechanism, and IoT devices can become a permanent problem, as proven by recent large-scale hacks. IoT security cameras employ heterogeneous, closed-source firmware focused on business but not security, whereas with the edge, resources are finite, and reusability is common. Manufacturers and developers should check their compliance preparedness when developing firmware and after it has been deployed. Companies may measure compliance preparedness to standards and regulations across sectors by conducting comprehensive firmware implementation analysis and monitoring.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] "New Security Signals study shows firmware attacks on the rise; here's how Microsoft is working to help eliminate this entire class of threats - Microsoft Security Blog." https://www.microsoft.com/security/blog/2021/03/30/new-security-signals-study-shows-firmware-attacks-on-the-rise-heres-how-microsoft-is-working-to-help-eliminate-this-entire-class-of-threats/ (accessed Apr. 07, 2023).

[2] "Gartner Report | Mitigate Firmware Risks in Data Centers | Eclypsium." https://info.eclypsium.com/gartner-mitigate-firmware-risks-in-data-centers (accessed Apr. 07, 2023).

[3] "NVD - Search and Statistics." https://nvd.nist.gov/vuln/search (accessed Apr. 07, 2023).

[4] Son M, Kim H. Blockchain-based secure firmware management system in IoT environment. Int Conf Adv Commun Technol ICACT Apr. 2019;vol. 2019-February: 142–6. https://doi.org/10.23919/ICACT.2019.8701959.

[5] Poonam Thakur;, Varsha Bodade;, Angitha Achary;, Madhuri Addagatla;, Neeraj Kumar;, and Yogesh Pingle, "Universal Firmware Upgrade Over-The-Air for IoT Devices with Security | IEEE Conference Publication | IEEE Xplore," 2019, Accessed: Apr. 07, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8991250.

[6] Zhu X, Li Q, Chen Z, Zhang G, Shan P. Research on security detection technology for internet of things terminal based on firmware code genes. IEEE Access 2020;8: 150226–41. https://doi.org/10.1109/ACCESS.2020.3017088.

[7] L. Zhu, X. Fu, Y. Yao, Y. Zhang, and H. Wang, "FIoT: Detecting the memory corruption in lightweight IoT device firmware," *Proc. - 2019 18th IEEE Int. Conf. Trust. Secur. Priv. Comput. Commun. IEEE Int. Conf. Big Data Sci. Eng. Trust. 2019*, pp. 248–255, Aug. 2019, doi: 10.1109/TRUSTCOM/BIGDATASE.2019.00041.

[8] M. Novak and P. Skryja, "Efficient partial firmware update for iot devices with lua scripting interface," *2019 29th Int. Conf. Radioelektronika, RADIOELEKTRONIKA 2019 - Microw. Radio Electron. Week, MAREW 2019*, Apr. 2019, doi: 10.1109/RADIOELEK.2019.8733437.

[9] C. Gao, L. Luo, Y. Zhang, B. Pearson, and X. Fu, "Microcontroller based IoT system firmware security: Case studies," *Proc. - IEEE Int. Conf. Ind. Internet Cloud, ICII 2019*, pp. 200–209, Nov. 2019, doi: 10.1109/ICII.2019.00045.

[10] Jang D, Kim T, Kim D. Dynamic analysis tool for IoT device. Int Conf ICT Converg Oct. 2020;vol. 2020-October:1864–7. https://doi.org/10.1109/ICTC49870.2020.9289204.

[11] J. Wang, H. Li, J. Ye, and J. Xiao, "Research on Intelligent Reverse Analysis Technology of Firmware of Internet of Things," *2021 IEEE Int. Conf. Power, Intell. Comput. Syst. ICPICS 2021*, pp. 164–169, Jul. 2021, doi: 10.1109/ICPICS52425.2021.9524146.

[12] Zandberg K, Schleiser K, Acosta F, Tschofenig H, Baccelli E. Secure firmware updates for constrained IoT devices using open standards: A reality check. IEEE Access 2019;7:71907–20. https://doi.org/10.1109/ACCESS.2019.2919760.

[13] M. H. Tsai, Y. C. Hsu, and N. W. Lo, "An Efficient Blockchain-based Firmware Update Framework for IoT Environment," *Proc. - 2020 15th Asia Jt. Conf. Inf. Secur. AsiaJCIS 2020*, pp. 121–127, Aug. 2020, doi: 10.1109/ASIAJCIS50894.2020.00030.

[14] Wang Y, Shen J, Lin J, Lou R. Staged method of code similarity analysis for firmware vulnerability detection. IEEE Access 2019;7:14171–85. https://doi.org/10.1109/ACCESS.2019.2893733.

[15] Z. Ahmed, I. Nadir, H. Mahmood, A. Hammad Akbar, and G. Asadullah Shah, "Identifying Mirai-Exploitable Vulnerabilities in IoT Firmware through Static Analysis," *1st Annu. Int. Conf. Cyber Warf. Secur. ICCWS 2020 - Proc.*, Oct. 2020, doi: 10.1109/ICCWS48432.2020.9292382.

[16] Zhang M, Zhou J, Cong P, Zhang G, Zhuo C, Hu S. LIAS: A lightweight incentive authentication scheme for forensic services in IoV. IEEE Transactions on Automation Science and Engineering April 2023;20(2):805–20. https://doi.org/10.1109/TASE.2022.3165174.

[17] Palmese F, Redondi AEC, Cesana M, "Feature-Sniffer: Enabling IoT Forensics in OpenWrt based Wi-Fi Access Points,",. IEEE 8th world forum on internet of things (WF-IoT). Yokohama, Japan 2022;2022:1–6. https://doi.org/10.1109/WF-IoT54382.2022.10152146.

[18] K. Sager, C. Fofie and A. Podhradsky, "Internet of Medical Things: Forensics Investigation on Zebra Phones," 2022 IEEE 19th International Conference on Mobile Ad Hoc and Smart Systems (MASS), Denver, CO, USA, 2022, pp. 736-741, doi: 10.1109/MASS56207.2022.00115.

[19] O. Almazrouei, P. Magalingam, M. Kamrul Hasan, M. Almehrzi and A. Alshamsi, "Penetration Testing for IoT Security: The Case Study of a Wireless IP Security CAM," 2023 IEEE 2nd International Conference on AI in Cybersecurity (ICAIC), Houston, TX, USA, 2023, pp. 1-5, doi: 10.1109/ICAIC57335.2023.10044176.

[20] S. Liu, X. Xu and Z. Nan, "Automated Behavior Identification of Home Security Camera Traffic," 2023 International Joint Conference on Neural Networks (IJCNN), Gold Coast, Australia, 2023, pp. 1-8, doi: 10.1109/IJCNN54540.2023.10191470.

[21] "D-Link." https://dlinkmea.com/index.php/product/details?det=ek1USzkyVUswTE5xZ0JsMEQwSjNTQT09 (accessed Apr. 07, 2023).

[22] "D-Link Technical Support." https://support.dlink.com/productinfo.aspx?m=dcs-5020l (accessed Apr. 07, 2023).

[23] "binwalk | Kali Linux Tools." https://www.kali.org/tools/binwalk/ (accessed Apr. 07, 2022).

[24] "What is LZMA Compression? - MajorGeeks." https://www.majorgeeks.com/content/page/what_is_lzma_compression.html (accessed Apr. 07, 2023).

[25] "GitHub - binvis/binvis.io: The binvis.io site." https://github.com/binvis/binvis.io (accessed Apr. 07, 2023).

[26] "'dd' command in Linux - GeeksforGeeks." https://www.geeksforgeeks.org/dd-command-linux/ (accessed Apr. 07, 2023).

[27] "How to Use the Strings Command on Linux." https://www.howtogeek.com/427805/how-to-use-the-strings-command-on-linux/ (accessed Apr. 07, 2023).

[28] "What is the Mirai Botnet? | Cloudflare." https://www.cloudflare.com/en-in/learning/ddos/glossary/mirai-botnet/ (accessed Mar. 29, 2023).

[29] "Uber Data Breach Affects 57 Million Rider and Driver Accounts." https://www.lifelock.com/learn/data-breaches/uber-data-breach-affects-57-million-rider-and-driver-accounts (accessed Apr. 07, 2023).