Reverse Engineering and Pwning the Router Firmware by Exploiting Cross-Site Scripting (XSS) or Weak Authentication (CVE-2020-24579)

IT19976686 – Jayasinghe D. A.
Faculty of Computing
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
dularaanushka17@gmail.com

***Abstract* - Throughout the years cyber-attacks have indeed evolved into the next-generation weaponry for gaining dominance or causing destruction in all aspects of everyday lives. Based on Cisco, attackers strive to profit financially through such exploits. Nearly more than 50 percent of these kinds of breaches cost around half of million dollars or even more. The discovery implements that numerous number of 0-day vulnerabilities can draws as well as encourages hackers to stealthily abuse such flaws. 0-day vulnerabilities such as coding defects, unfixed or even obsolete programs, and holes in equipment firmware. Assault against this sort of program is difficult to identify, and the diagnosis probability is extremely weak. Yet when manufacturers issue firmware upgrades or fixes, just the corporate layer is informed and notified of the modifications. Through the advancement of technologies including the rise of the Internet of Things, the majority of domestic products are now networked to the internet, and people access to the internet through Wi-Fi routers. Users do not desire toward being conscious of the underlying procedure of current routers because of the user-friendliness of such equipment. Because of this obliviousness, several exploits were launched, as well as hackers were able to execute Man-In-The-Middle attacks potentially seize hold over the infrastructure. The focal point of this document is to describe the backdoor risks and inadequate authentication exploits performed done by reverse-engineering the software/firmware preinstalled in Wi-Fi routers.**

***Keywords* - 0-day vulnerabilities, Internet of Things, router firmware, Wi-Fi router, reverse engineering, backdoor**

I. INTRODUCTION

On the one hand, advanced technology facilitates employment, but on the other hand, complex innovation facilitates cyber-attacks. Cyberattacks are growing in tandem with technological advancements. As a result, the zero-day vulnerabilities, and Google's "Project Zero" treats them like a bug bounty program. Routers, along the other hand, are connected to the internet by practically every equipment, and therefore are often overlooked or neglected in exploitation studies. Because most router firmware hacks are zero-day hacks, they are difficult to spot. Such activities carried out by reverse engineering and pwning router firmware in Linux settings using multiple techniques.

II. RESEARCH SATEMENT

Its major topic of this research project is pwning the router firmware and gaining access to the reverse shell. Moreover addition, the article discusses CVE-2020-24579 related insecure verification in network firmware.

III. OVERVIEW OF REVERSE ENGINEERING

A. Reverse Engineering

This practice of disassembling a unit and seeing how it functions in terms of creating or enhance it is known as "reverse engineering" or "back engineering". The approach, that dates back to the industrial revolution, is currently widely employed in computer systems. The technique of getting a program's source code is known as reverse engineering software. The majority of network software is contained in binary code. Binary formatting alludes to the saving of data as 1's and 0's, or even in binaries. Because binaries structured information is not readable and understandable, it must be reverse engineered then decrypted in order to get documents inside the operating system.

Following that, reverse engineers evaluate these documents to determine the patterns as well as scripting problems. Backdoors typically introduced within the software using the bugs discovered during the analysis step. Whenever an intruder compromises a gateway in a system, every units linked to that router are likewise hacked, giving the attacker complete control over the system. Reverse engineering is a method of modifying a product. Reverse engineering is a fascinating way to learn about the core mechanism of a technology, including both benefits and drawbacks.

B. Firmware

A firmware is a software application or a group of tasks that is installed on a machine. Read-Only Memory contains encoded commands. Because ROM is a non-volatile storage and a flash memory, it could be rewritten but firmware cannot be deleted by the users. Software could now be discovered on almost every electrical equipment, including cellphones, TVs, freezers, and routers. System firmware functions as a bridge in between client and the equipment, simplifying the equipment's sophistication. The configuration for the routers is centered on the Linux operating system [4].

C. Weak Authentication

"Authentication" relates to the procedure of authenticating person's identity to the an program particular equipment. Single-Factor Authentication (Primary Authentication), Two-Factor Authentication (2FA), Single Sign-On (SSO), Multi-Factor Authentication (MFA), and more verification methods are available. However, the Credential Authentication Process (Password Authentication) is by far the most widely used authentication method. We employ every one of these digital certificates to safeguard a significant item; nevertheless, inadequate verification occurs when the quality of the authentication mechanisms is somewhat insufficient in contrast to the safeguarded value. CVE-2020-24579 is a vulnerability in routers software related to weak authentication. Unverified users can gain entry to certified or protected sites by launching an attack. Unprotected login can be caused by a variety of factors, including credential rules, passwords complexity, and credential reuse.

D. Tools and Methodologies for Reverse Engineering

Reconstruction and restoration, also known including forward development, is the method of constructing application specific on the needs of customers. The phrases reverse engineering, restructuring, and re-engineering are used in reverse engineering forward designing. Inspection, analysis, and study of which was before software is the most prevalent and official approach of reverse engineering. Binwalk, Hexdump, Objdump, Strings, Radare2, and other software engineering toolkits. Ghidra is also an open-source reverse engineering tool built and maintained by the National Security Agency (NSA). Ghidra is currently at version 9.2.4. Various reverse engineering software, as well as their explanations, were mentioned beneath [3].

a) Binwalk, is really a program which searches for preinstalled data files and executable programs in an immersing manner. Binwalk is a program that analyzes software binaray packages. In binwalk, the central library became Libmagic.

b) Objdump, which collects details about entity type documents. It also presents all of the gathered data following evaluating and sorting it. Objdump is well-known among developers who use compilation techniques.

c) Strings, which are utilized to retrieve information from executable documents in string form.

d) GDB, is a well-known debugging that can decompile and dismantle binary and application items created in the C and C++ programming languages.

e) Unsquashfs, also known as the squashed operating system, is a compacted and read-only file system that runs on Linux. It compresses data in blocks that range from 4K to 64K. Unquashfs is a program that can decompress squash files.

f) Radare2, a tool designed specifically for reverse engineering. The majority of firmware packages are in binary code, and Radare2 can examine source code.

g) IDA, this professional standard program is utilized for debug objectives when revrse engineering is done at the corporate level. IDA is also used to retrieve code for application programs.

IV. FILE SYSTEM

The manner of an information can be stored and retrieved is determined by the firmware's or program's data format. The two most used data structures in the Windows platform are New Technology File System (NTFS) as well as File Allocation Table (FAT). SquashFS (Squash File System) and UBIFS (Unsorted Block Image File System) are two of the most famous file systems found in Linux distros. SquashFS (.squafs) is a read-only compacted storage device, based on Linux that has been

utilized in routers firmware. Block sizes ranging from 4KB to 1MB are also permitted. OpenWRT is a Lempel-Ziv-Markov Algorithm (LZMA) condensed open-source operating system designed for small devices.

A. *ELF and MIPS Architecture*

Microprocessor without Interlocked Pipeline Stages, or MIPS design. MIPS has evolved into five types since its initial launch. Reduced Instruction Set Compiler (RISC) is the foundation of MIPS. In MIPS, there are two structures: 32-bit and 64-bit. Microprocessors employ MIPS as its embedded system. The Compiled code and Linkable Format is a standard for executable files that is widely used. It handles both little and big endianness, as well as a variety of memory locations.

V. Backdooring Overview

A. *Backdooring Process*

Throughout this assault, the assailant initially performs the backdooring step, which allows the assailant or attacker using the intended routers as a bot. “Pwning” is another term for this procedure. An illegal individual can capture the information of the administrator in the router's login form by exploiting a Cross Site Scripting (XSS) weakness or taking full abuse of a router's insufficient protection. The protection of the routers' frontend web sites has been strengthened as innovation has advanced. These backdoor assaults are difficult to detect since the user of something like the infected device must examine each word of the software code base for defects or changes. Because complete reverse technology and quality is necessary for this technique, none of us can identify a corrupted router. The backdooring procedure is presented in Figure 1 [6].

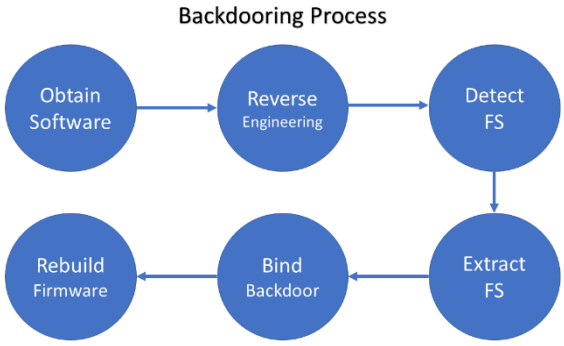

Figure: 1: Backdooring Process

VI. Reverse Engineering And Backdoor Attack Analysis

A. *Obtaining the Firmware and Detecting the file system*

The firmware file can be downloaded at www.tp-link.com, the authorized TP-Link webpage. TP-Link is a well-known Wi-Fi device company. The software versions can be selected from an options list on the webpage; for this procedure, version 8 is being used. Software for the TL-WR843N V8 140734 can be obtained as a zip format by selecting the data file. The software that was obtained is for thier TL-WR841N router variant. Once you unpack the uploaded zip archive, you'll find two documents, one of whom is a pdf with instructions for updating the router. The intended software would be the other package. The data file '.bin' can be used to identify it. The extension '.bin' denotes compacted binary files, that are not in clear text and are in a ciphertext for humans. As illustrated in the image beneath, the function 'file' displays the firmware version, sequence number, hostname, and size of the file in bytes.

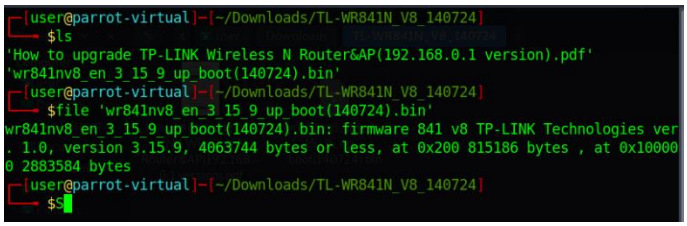

Figure: 2: Firmware Details

B. *Extracting the file information using Binwalk*

Binwalk is a Kali-Tool for extracting data from file types. This can recognize hidden representing a total and folders within bios files. Binwalk is linked to Libmagic Components. Binwalk contains data thus the kernel point of entry default location, kernel offset, rootfs compensate, and bootloader counteracted. The intended software compresses data using the Lempel-Ziv-Markov Algorithm (LZMA). The system utilizes the Squashfs storage device. Endianness refers to the byte arrangement in which data is stored. Little-endian and big-endian are also the two sorts. Little-endian methods store information from the least significant bit, while big-endian methods store values from the most significant digit. The big-endian technique is used by the TP-Link software. As a result, the attack code should be written in the very same big-endian structure as that of the bios [5].

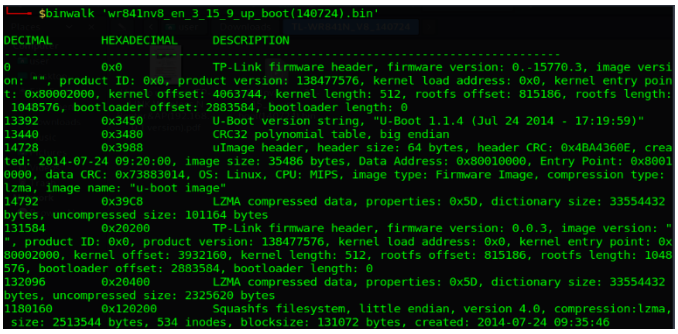

Figure: 3: Binwalk Details

C. *Extracting the firmware*

The utility 'Firmware-mod-kit' is used to retrieve any software. This is an open-source implementation accessible from github that can be freely cloned as well as integrated through into server. Parrot OS and Firmware-mod-kit are extremely friendly (Parrot OS 14.11). The intended software is now in the same location as the utility in order to retrieve it. The firmware is extracted using the "./extract-firmware.sh" function. The kernel is extracted within the 'fmk' directory found in the firmware-mod-kit directory. There will be three subcategories in the 'fmk' file: 'image partss',logs,' rootfs.' The 'rootfs' directory contains the whole documents relevant to the firmware.

Figure: 4: Firmware Extraction

D. *Generating the payload*

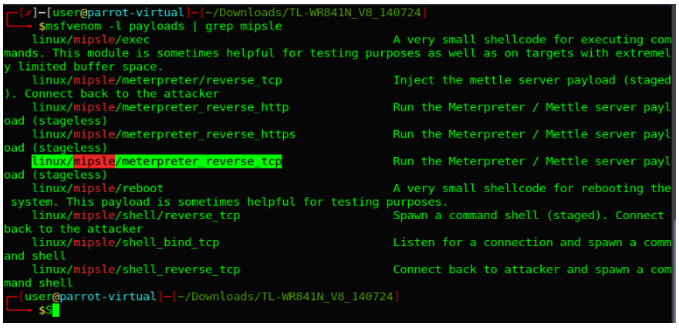

Figure: 5: Generating the payload

The software structure details is contained within 'Busybox file.' The bios is written in the "elf " runtime style and runs on a 32-bit MIPS 32 platform. As a result, the payloads that attack the system should be built in the equivalent design. Due to the obvious concerns with fixed library, two manually payloads methodologies that can be used are buildroot and bindshell. The Metaslpoit framework is utilized. The Metasploit infrastructure can be used to generate payloads automatically. The function "msfvenom -l | grep-mipsle" had been used to build the corresponding payloads, as seen in the accompanying image.

The payload "meterpreter reverse-tcp" was chosen from of the created list and stored inside the " etc/init.d/ " location for attachment to the reverse-shell.

VII. Reconstruction of Firmware and Handler Setup

A. *Reconstruction of Firmware*

Due to problems, routers reboot routinely, and users shut down devices while users were not under operation. As a result, the payload should resume after every reboot in order for such pwning operation to continue. The optimal answer for auto-starting the injection is to store the payload directory structure in the rcS file that runs the init scripts. BIOS development functionality was formerly provided in Firmware-mod-kit. The command of " ./build-firmware.sh fmk" was used to rebuilding the firmware, as shown in the diagram following.

Figure: 6: Firmware Reconstruction

B. *ASLR, DEP and Stack Canaries*

The majority of firmware or equipment makers do not address the protection of their firmware at the compiled code. Firmware depending on MIPS, ELF, and ARM-based firmware, as according to sources on the internet, lacks stacks and source code encryption. The Addressing Space Layout Randomization (ASLR) function is being used to safeguard the stacks from hostile payloads as well as to avoid buffer overflow problems [7]. Assailants will be unable to connect payload towards the source code whenever ASLR is applied during generating the firmware. Data Execution Prevention (DEP) is a Windows privacy solution that ensures the protection of memory accesses and storage addresses against destructive payloads connected to firmware. DEP changes entire data segments in memory locations to a non-executable form. Returning addresses are used in the codebase of all programs,

and sometimes intruders could use them to change the memory location. Stack canaries defend the stack against a variety of threats. Stack canaries are used by developers throughout the formulation phase. However, the firmware lacks these fundamental security features. Disregarding those safeguards and computers engineers' recklessness resulted in catastrophic harm to the entire it's architecture where such a firmware was utilized [8].

C. *Configuring up the handler*

Perhaps there is a listening connection towards the backdoor once it has been established. Whenever the compromised firmware is loaded into in the intended device then rebooted, the backdoor inside the rcS directory starts looking for the specified Internet Protocol address with LHOST of the preconfigured Metasploit payload. The target network then uses that LPORT to establish communication to LHOST. Thorough active analyzation of the connections from the port is necessary for an effective assault. The victim's Local Internet Protocol address was used as the input again for LHOST and LPORT or the relevant port retrieved from target while generating the payloads with Metasploit.

VIII. FIRMWARE FLASHING

A. *Flashing the router with exploited firmware*

Rebuilt firmware is saved as " new-firmware.bin " in the fmk subdirectory. This must be installed or flashed through into routers as the following phase. Although devices have the capability to upgrade their firmware, modifications to essential features necessitate administrative access. This process is made easy by weak authentication. Attackers could gain access to the control configuration by exploiting inadequate authenticate weaknesses. CVE-2020-24579 is a routers problem linked to poor security. When an intruder gained control to the administrator privileges through any manner, the compromised software can be uploaded. The payload then establishes the link here between intruder as well as the router.

Figure: 7: Flashing Firmware

B. *Router Pwned*

The router requires approximately 5 to 10 seconds to reboot after just a complete firmware upgrade, as well as the rcS script with the payloads requires about 50 seconds to establish the link. The LPORT will provide the link to Metasploit if the attack takes place. Lastly, because a routers is a crucial element of a system, an intruder could seize complete control system. An intruder can send orders to the overall infrastructure from afar. If somehow the hacked router is connected of an IoT ecosystem, the intruder can immediately manipulate the devices connected to it. Moreover, the assailant can use this assault to carry out a variety of many other attacks, such as launching second strike against a carefully designated machine linked to the router.

```
msf exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.86.223:4444
[*] Sending stage (179779 bytes) to 192.168.86.61
[*] Meterpreter session 1 opened (192.168.86.223:4444
 2018-05-29 11:48:32 -0400

meterpreter > shell
Process 3028 created.
```

Figure: 8: Reverse Shell

C. *Final Results*

If somehow the assault is succeeding, anti-virus other anti-malware software will be unable to identify any malicious code. As a consequence, the implanted backdoor gets filled every time the hacked device starts up, even without consent of the owner. Moreover, an intruder can construct a fake SSID to login into to the routers and allow entry towards the user's PC, after which the assailant may attach a keylogger to steal the user 's password. Unless the intruder gets past that point, he can get into enterprise networks.

IX. CONCLUSION

Assaults on routers firmware are hard to trace, and reverse engineering skills are required to spot changes in the programming code. Analyzing the firmware source code is a time-consuming and difficult operation. Such exploits are simpler since routers; customers don't really upgrade the software on a constant schedule or are ignorant of the upgrades. Furthermore, attackers are more likely to abuse routers sites with weak authentications. Through targeting such weak security flaws, hackers could upload the firmware as a firmware upgrade and made the victims to upgrade the router's firmware. The weakness CVE-2020-24579 was discovered on routers login methods. Assailants has to have a thorough understanding of the file systems available in the routers as well as pwning tools and techniques. Backdooring tools include binwalk, unsquashfs, objdump, strings, and GDB. The Metasploit framework also includes a large number of payloads. A reversed tcp payload was employed in this attack. The intruder may listen actively to the link established by that payload following establishing the

listener. Since the payload is inside the rcS path, it is regularly initiated after every router startup. The hijacked device can be used to execute a variety of additional assaults. Additionally, cyber-attacks might affect devices linked towards the targeted device.

ACKNOWLEDGEMENT

Dr. Lakmal Rupasinghe and Ms. Chethana Liyanapathirana of the Sri Lanka Institute of Information Technology's Faculty of Computing, whose experience, understanding, advice, and unwavering support allowed us to work on this project. I would also like to thank my friends for providing me with the motivation to succeed in my work. I am grateful to every one of my family members for believing with me and encouraging me to achieve my life goals.

REFRENCES

[1]2021. [Online]. Available: https://owasp.org/www-chapter-coimbatore/assets/files/Router%20Reversing%20by%20Adithyan%20AK.pdf. [Accessed: 10- May- 2022].

[2]"Reverse Analysis and Vulnerability Detection for Network System Software", *Ieeexplore.ieee.org*, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/5207900/. [Accessed: 10- May- 2022].

[3]"Reverse engineering attacks: 6 tools your team needs to know | TechBeacon", *TechBeacon*, 2021. [Online]. Available: https://techbeacon.com/security/reverse-engineering-attacks-6-tools-your-team-needs-know. [Accessed: 12- May- 2021].

[4]"Firmware Definition", *Techterms.com*, 2021. [Online]. Available: https://techterms.com/definition/firmware. [Accessed: 13- May- 2021].

[5] *Tools.kali.org*, 2021. [Online]. Available: https://tools.kali.org/forensics/binwalk. [Accessed: 14- May- 2021].

[6]"Offensive IoT Exploitation Exam – Backdooring a firmware – The sh3llc0d3r's blog", *Sh3llc0d3r.com*, 2021. [Online]. Available: https://sh3llc0d3r.com/iot-backdooring-a-firmware/. [Accessed: 14- May- 2021].

[7]"On the effectiveness of DEP and ASLR – Microsoft Security Response Center", *Msrc-blog.microsoft.com*, 2021. [Online]. Available: https://msrc-blog.microsoft.com/2010/12/08/on-the-effectiveness-of-dep-and-aslr/. [Accessed: 14- May- 2021].

[8]"Most popular home routers lack basic software security features - Help Net Security", *Help Net Security*, 2021. [Online]. Available: https://www.helpnetsecurity.com/2019/01/07/home-routers-software-security/. [Accessed: 15- May- 2021].

AUTHOR PROFILE

Dulara Anushka is a hardworking and ambitious undergraduate at the Sri Lanka Institute of Information Technology (SLIIT). He enrolled at SLIIT to pursue a Bachelor of Science (Hons) degree in Information Technology Specializing in Cyber Security after passing his OL and AL Exams. He also belonged to the school's Technical Association, Robotics Association, and ICT club. He is presently in his third year of studies and works at Millennium IT ESP in Sri Lanka as an Intern – Cyber Security.

While fulfilling the academic tasks, he acquired certificates of Cybrary Zero Trust Networks, CISCO Introduction to Cybersecurity, and CISCO Cybersecurity Essential. He has strong skills to work with virtualization, Kali Linux OS, Parrot OS, and Nessus, Burp Suite, and many other security tools. He is proficient to work with the Linux environment, programming languages: C and C++, web technologies: HTML and CSS, and is familiar with Microsoft Office 356.

CVE-2019-14287 Vulnerability Exploitation, Web Audit, and Automated Parking System Website are just a few of his undergraduate work. His abilities aren't limited to academic work. He has worked as a team leader in several group initiatives due to his strong communication and leadership abilities.