Security Attack on IoT Related Devices Using Raspberry Pi and Kali Linux

Batrisyia B Mohd Bakry
School of Electrical Engineering
College of Engineering
Universiti Teknologi MARA
Shah Alam, Selangor, Malaysia
batrisyia.mbakry@gmail.com

Alisa Rafiqah Bt Adenan
School of Electrical Engineering
College of Engineering
Universiti Teknologi MARA
Shah Alam, Selangor, Malaysia
alisarafiqah25@gmail.com

*Yusnani Bt Mohd Yussoff
School of Electrical Engineering
College of Engineering
Universiti Teknologi MARA
Shah Alam, Selangor, Malaysia
yusna233@uitm.edu.my

***Abstract* – Currently, the growth of cybercrime and online attacks reflects the increase in the use of Internet of Things (IoT) devices. This has become a challenge as the use of IoT devices continues to grow and diversify in variety. It is challenging to cater to all security problems at once due to IoT devices have many designs and updates. These variations are likely to create security concerns, as there is currently no industry-wide standard for IoT device security. This situation raises security concerns. In addition, since IoT devices hold sensitive and personal information, they are susceptible to a variety of risks as attackers attempt to take advantage of the flaws that exist. As a consequence, ensuring the safety of all applications and devices connected to the IoT is crucial. This paper presents a project that executes penetration testing on a Raspberry Pi 4 that could act as an IoT device and a laptop running on Kali Linux as a hacking tool in order to determine the vulnerabilities of IoT devices. This project successfully conducted cyber-attacks against the exploitable vulnerabilities in the Raspberry Pi 4 acting as an IoT device by executing DoS attack, man-in-the-middle attack and brute force attack. Finally, this paper also presents a set of best practices for mitigating the cyber-attacks.**

Keywords—cybersecurity, penetration testing, vulnerabilities, Raspberry Pi 4

I. INTRODUCTION

A. Overview of Study

The Internet of Things (IoT) usage has arisen in the past few years as technology keeps evolving and advancing [1]. IoT technology is diverse as it can be applied to almost any technology capable of exchanging data over the Internet [2]. Now, as Wi-Fi and cellular data connections are becoming more widespread, more and more everyday devices are being equipped with the ability to connect to the Internet. IoT gives endless possibilities [2]. IoT devices such as drones, smart locks, smart bulbs, smart fridge, and many more IoT devices could be great helps in our daily lives. However, the growing use of IoT devices causes more cyberattacks. IoT devices contain sensitive and private data, exposing them to many threats that aim to take advantage of these weaknesses, which are the current IoT infrastructures [3]. Numerous attacks, such as Distributed Denial-of-Service (DDoS) attacks, eavesdropping on people's daily activities, and taking over communication links to seize control of remotely controlled objects, have been attempted against IoT devices [3].

These security problems result in a considerable impact, especially in industries with no tolerance for security breaches. Due to the increasing number of IoT devices, the security issue has become a challenge. While tools can assist in analysing software for vulnerabilities before release, the diverse platforms make scanning challenging and demand technological development [4, 5]. Although much research has been conducted to address these concerns, a common standard for IoT devices has still not been established [6, 7].

Security has become crucial for IoT devices, and studies and surveys have been conducted on the vulnerabilities in IoT devices. For example, Andrei Costin [8] has reviewed the existing threats and vulnerabilities in video surveillance, closed circuit TV and IP-camera systems based on publicly available data and chose seven measures to describe attacks and mitigate these attacks. However, the review only discussed the monitoring devices while various types of IoT devices exist. Not only that, it also does not focus on vulnerability detection. Next, [9] provided an overview of the vulnerabilities found in embedded system devices. They categorized existing methods in firmware vulnerability detection as fuzzy testing, homology analysis, behavioural analysis, and symbolic execution. Nevertheless, there are also particularly significant works outside these classifications.

[10] published a paper on the vulnerabilities in Public Wi-Fi and how to mitigate these attacks by performing a test on public Wi-Fi security using Raspberry Pi and Kali Linux by executing attacks such as Domain Name System (DNS) Spoofing, Wi-Fi password cracking, Man-in-the-Middle, and Evil Twin. However, the paper focuses only on the vulnerabilities of public Wi-Fi, not IoT devices. [11] launched a DoS attack on an IoT system. The attack tool is Kali Linux and uses 3 DoS methods on an IoT system. The experiment results are used to analyse and compare the effects of these three DoS methods. On the other hand, [12] also compared three types of DoS attacks on IoT devices using Kali Linux as the attacker and Arduino as the victim. However, these two papers only focus only on one kind of attack, which is the DoS attack.

This project aims to assess the vulnerabilities of IoT devices such as drone, smart lock, smart bulb, or many other IoT devices through conducting penetration testing on a Raspberry Pi 4 with the hacking tool Kali Linux. Installing an operating system and configuring the Raspberry Pi to connect to the Internet makes itself vulnerable to cyber-attacks. These attacks, such as Man-in-the-Middle (MITM), DoS, and Brute Force, are performed against the Raspberry Pi 4 using Kali Linux to look for exploitable vulnerabilities. These threats and vulnerabilities are evaluated to suggest a set of best practices for mitigating cyber-attacks.

The following section of this paper is organized as follows: Section II will discuss the method used to execute the penetration testing on the Raspberry Pi, Section III will discuss the result obtained and the best practice to mitigate cyber-attacks, and Section IV contains the conclusion of the paper and is followed by the reference list.

2022 International Conference on Computer and Drone Applications (IConDA) | 978-1-6654-9235-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICONDA56696.2022.10000370

II. METHODOLOGY

This section will show how each attack on the Raspberry Pi is executed. There will be three types of attacks that will be simulated, which are:

1. Brute Force Attack
2. MiTM Attack
3. DoS Attack

Each attack is executed using different tools in Kali Linux, such as Nmap, Bettercap, and Xerosploit. The Raspberry Pi 4 will act as an IoT device, which is the victim of the cyber-attacks, and the Kali Linux as the attacker.

The first step of penetration testing is to set up a testbed, as seen in Fig. 1, which consists of a computer running with a Kali Linux Operating System acting as the attacker, a wireless adapter to capture wireless traffic, and a Raspberry Pi 4 with a Raspberry Pi Operating System acting as the victim.

Fig. 1. Testbed Implementation

Fig.2 illustrates the flowchart for the overall process of all three attacks done for the penetration testing on the Raspberry Pi 4 using Kali Linux.

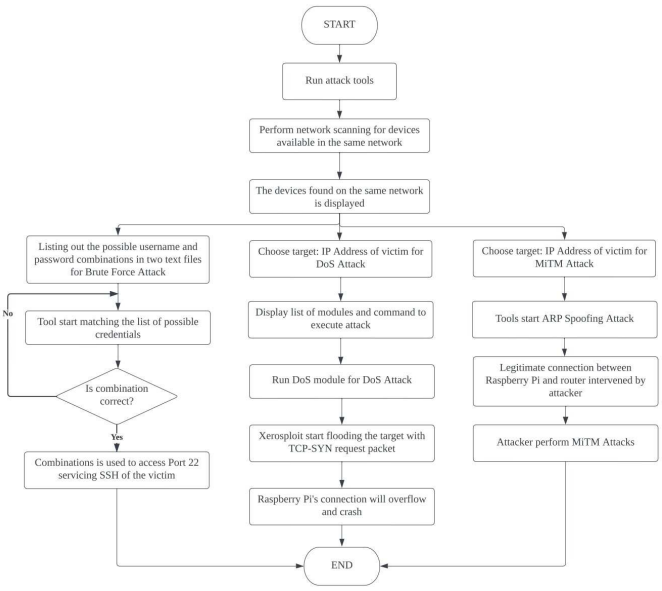

Fig. 2. Flowchart of Penetration Testing on Raspberry Pi

A. Brute Force Attack

Brute Force Attack is an attack where hackers use a set of password and username combinations to guess login information. Brute Force Attacks can also be used on IoT devices, allowing hackers to access the system and navigate freely within it. As a result, they can either steal valuable data or spread malware into the victim's system, causing disturbances or conducting cybercrimes. Brute Force Attacks is an ancient attack method that inexperienced hackers primarily utilize. However, they are still popular among hackers and have been proven effective. Depending on the length and complexity of the password, breaking the password can take anything from a few seconds to many years.

Fig. 3 shows the process of how this attack is done on the victim. Two different tools are used to execute this attack: Nmap and Hydra. Nmap is a network mapper and is one of the most popular networks scanning tools. Network administrators usually use it to map their networks. This tool can find live hosts on a network, scanning ports, version detection, Operating System detection, and many more. On the other hand, Hydra is a tool that solely focuses on brute-force attacks where it supports different types of protocols to attack, such as FTP, SSH, Telnet, and MS-SQL.

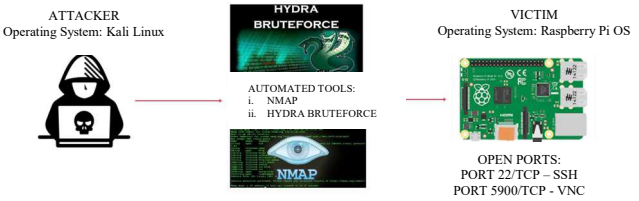

Fig. 3. Process of Brute Force Attack

These tools, which are Nmap and Hydra, will start scanning for devices in the same network to decide the target they want to attack. Once the scanning is complete, it will display the devices found on the same network as the attacker. These tools will provide not only the IP addresses of these devices but also the ports and services that are either open or close. This is the point at which the attacker can collect information about the target, which in this example is the Raspberry Pi with port 22 open. Port 22 is known to support Secure Shell (SSH), a protocol that enables another computer or device to control another computer or device remotely, and attackers would attempt to exploit this open port.

The next step is to list down the possible username and password combination in two different text files, as seen in Fig. 4. These two text files will be called in the commands to execute the brute-force attack. To find the correct username and password combination, the attacker will run the **command Nmap 172.20.10.4 -p 22 --script ssh-brute -script-args userdb-username, passwdb-password** on Nmap. However, if the attacker uses Hydra, the command would be **Hydra -L username -P password ssh://172.20.10.4 -t 8**. Then, the tools would start matching the username and password combination to obtain the correct login credentials.

Fig. 4. Possible Username and Password Combinations

Once the Kali Linux tools manage to identify the correct combination of username and password, the corresponding tools will display a clear text of the credentials obtained as shown in Fig. 5 and Fig. 6. The attacker can now use this information to access the victim's device through the SSH port and be able to navigate freely into the victim's device.

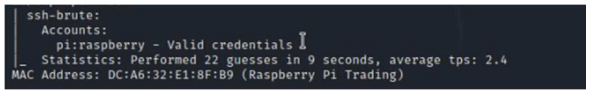

Fig. 5. Nmap successfully gain the login credentials to the victim's device

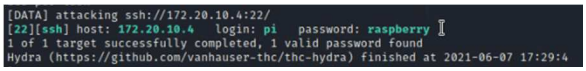

Fig. 6. Hydra successfully gain the login credentials to the victim's device

B. Denial-of-Service (DoS) Attack

DoS attack is an attack that shuts down a computer or network that could prevent users from using these devices. All services are denied during this attack because it floods the target with traffic or sends information until it triggers a crash. However, a DoS attack does not result in theft or loss of data, but it may cost the victim a great deal of time and money to handle.

As seen in Fig. 7, the attacker will execute a DoS attack on the targeted device where it uses flood attacks that are called SYN flood. This attack works by exploiting the handshake process of a Transmission Control Protocol (TCP) connection. For this attack, the tool used is Xerosploit, which is a penetration testing toolkit. Like Nmap, Xerosploit can scan the devices available in the same network as the attacker, and they will choose their target to launch these attacks.

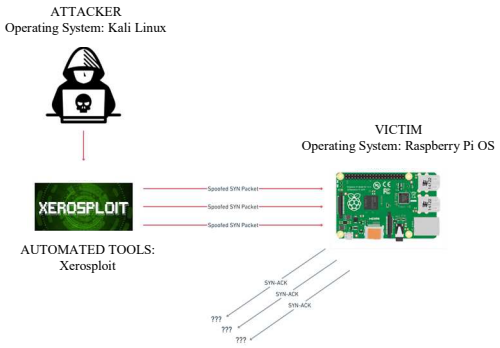

Fig. 7. Process of DoS Attack

The attacker will begin by selecting a victim. It was the Raspberry Pi with the IP address 172.20.10.4 in this instance. The attacker can now perform any attack from the list of available modules in Xerosploit on the targeted device. The command run is applied to execute the DoS attack, and Xerosploit will start flooding the target with the TCP-SYN request packet. This module uses the HPING tool that continuously sends repeated SYN packets to every targeted port, which causes the target to receive multiple requests to establish communication. The target will respond to each request with an SYN-ACK packet from each open port. However, the attacker will not reply to the target with the expected ACK and leave the target to wait as it cannot close the connection without getting replied. Once the connection has timed out, another SYN packet will arrive. The target's connection will eventually overflow, and service to legitimate clients will be denied, causing the target to malfunction or crash.

C. Man-in-The-Middle Attack

MiTM attack is where the attacker intervenes in the communication between two devices without the users' knowledge. MiTM attack is usually conducted with the intention to gain access to confidential information that may be useful to the attacker or cause disturbance to the victims.

This attack exploits the opportunities in Layer 2 or also known as the MAC layer, where it abuses the Address Resolution Protocol (ARP). Man-in-the-Middle attack on the Raspberry Pi is carried out using Bettercap and Xerosploit, as shown in Fig. 8.

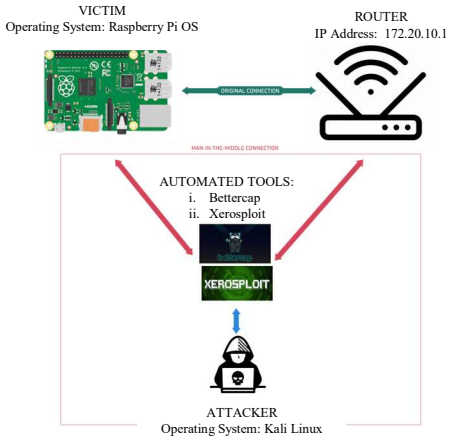

Fig. 8. Process of MiTM Attack

The tools, which are Xerosploit and Bettercap, will start by scanning the accessible devices by sending dummy User Datagram Protocol (UDP) packets to all available IP addresses on the subnet. The attacker's MiTM attack is conducted against the Raspberry Pi with the IP address 172.20.10.4. The attacker will execute an ARP Spoofing attack on the target in order to intervene in the connection between the victim and the router. The attacker will trick the victim and the router by replacing the MAC addresses of the victim and router with the attacker's MAC address. The victim will instead communicate with the attacker, thinking it is the router. On the other hand, the router will reply to the attacker, thinking it is the client. Once the attacker starts the ARP Spoofing Attack, the MAC address table of the victim's device will be altered to that of the attacker's MAC address.

Both the victim and the router will communicate via the attacker rather than directly with one another, allowing the attacker to sniff the packets they exchange. If the attacker uses Xerosploit, more advanced attacks can be executed than when using Bettercap. When victims visit an unprotected website, they can also see their web browser rattled and visuals altered by Xerosploit. In Xerosploit, the module **move** causes the web browser on the victim's device to shake. When this attack is launched, the tool will begin injecting JavaScript code into the target's browser whenever the victim visits an unsecured HTTP webpage. Another method of attack is to use the **replace** module, which replaces the images loaded on the victim's HTTP-based website with any image selected by the attacker. When the victim accesses an HTTP website, they will notice that the pictures have been replaced.

III. Result and Analysis

The objective of this project has been successfully achieved, which is to analyze the vulnerabilities in the Raspberry Pi 4 by performing penetration testing using the three attack methods.

Three attacks were successfully executed on the Raspberry Pi. Each attack uses different tools to be launched. This section

will show the results from the attacks and how to mitigate these attacks to minimize the impact.

A. *Brute Force Attack*

Through the credentials obtained from employing either of the two methods, Nmap and Hydra, the attacker can now access the victim's device through SSH. As a result, the attacker can freely roam through the victim's device remotely and execute operations such as file and folder transfers, disk utilization reviews, and performance monitoring using commands. In addition, the attacker can extract the victim's private information, install malware, and change the password, so the victim is unable to use their device. As seen in Fig. 9, the attacker has successfully accessed the victim's device through SSH.

Fig. 9. Attacker Accessing Victim's Device Through SSH

Fig. 10 shows the attacker successfully changing the password of the victim's device via SSH. When victims log onto their device, they are unable to access it as the attacker has successfully changed the password, as seen in Fig. 11. This is because most new IoT devices tend to have vulnerable default credentials. In this case, the username and password of the Raspberry Pi by default are pi and raspberry. Users that do not change this password are more vulnerable to attacks as these credentials are easily exploitable. On top of that, Port 22, which services SSH is open, can be easily hacked into if the victim uses a default or weak credentials that are easy to guess. In most cases, users usually establish weak passwords and frequently use the same password for many accounts. Weak and insecure passwords have long been a source of security concern, as they provide numerous advantages to attackers. Therefore, the user should become habituated to using a stronger and unique password for each account. However, employing private key authentication rather than passwords is preferable because the security risk is negligible.

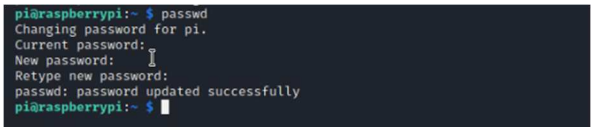

Fig. 10. Attacker Changing the Password of the Victim's Device

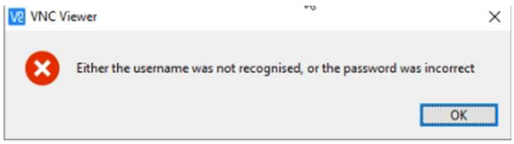

Fig. 11. Victims Fail Attempt to Access Their Device.

B. *Denial-of-Service (DoS) Attack*

The second attack that was executed on the Raspberry Pi was DoS Attack using Xerosploit. The DoS attack was the easiest to implement out of the three attacks. It usually does not give any direct monetary advantages to the attacker as this attack is only meant to shut down a machine or network, making it inaccessible for the victims. This DoS attack uses an SYN flood attack where the attacker sends SYN packets repeatedly to the target, and the target will try to reply to every request. This overwhelming traffic caused by the attacker will result in the target malfunctioning and crashing, thus denying services to the intended users.

Fig. 12 depicts the victim's device prior to the DoS attack when it can connect to the Internet without difficulty. At the same time, Fig. 13 displays the victim's device following the DoS attack when they are unable to surf the Internet due to the overwhelming SYN flood attack. This DoS attack employs an SYN flood approach, in which the attacker continually sends SYN packets to the victim, which attempts to respond to each request. The attacker's flood of traffic will drive the target to malfunction and crash, preventing the intended users from receiving any services.

Fig. 12. Before DoS Attack

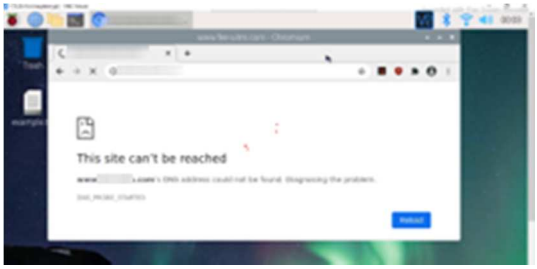

Fig. 13. After DoS Attack

DoS attacks are frequently directed at the web servers of businesses or organizations, particularly those in the financial, e-commerce and media, government, and trade domains. This is because, if the attack is carried out, the businesses will suffer financial loss due to their inability to supply their services to their clients. As a result, this attack can potentially bring down enterprises worth hundreds to millions of dollars.

Several measures must be taken into consideration to prevent this attack. For companies or organizations, having a response plan towards DoS or DDoS attacks is a must. The companies must assess their systems and identify potential security flaws, as well as the countermeasures to take in the event of an attack. Another prevention is by installing appropriate protection tools in the networks and applications such as firewalls, network monitoring software, anti-virus, anti-malware programs, and threat monitoring systems. Companies can monitor their network traffic to identify any changes. Maintain an up-to-date inventory of tools and methods, as attackers become more innovative as time passes and will constantly come up with new tools or methods to execute attacks. Systems should be updated frequently to ensure that any flaws or concerns are addressed, preventing attackers from exploiting these vulnerabilities. Early detection of risks is the most effective method of preventing cyberattacks.

C. *Man-in-The-Middle (MiTM) Attack*

The last attack that was executed on the Raspberry Pi was the MiTM attack. MiTM attack is an attack where it gains private information and can cause malicious activities on the victim's device. There are many approaches in starting a MiTM attack which are IP Spoofing, ARP Spoofing, and DNS Spoofing. However, for this project, the attack takes advantage of the ARP by using ARP Spoofing attack.

ARP spoofing is basically replacing the router's MAC address with the attackers in the victim's device and vice versa. As a result, both the router and victim will exchange data through the attacker allowing the attacker to obtain private information and cause disturbance to the victim. Fig. 14 shows the MAC Address table of the victim's device before the MiTM attack is executed, and Fig.15 shows the MAC Address table is altered after the attack.

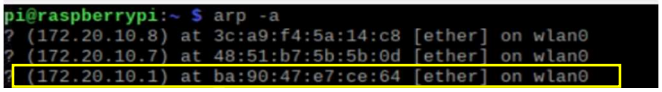

Fig. 14. MAC Address table in the victim's device before the MiTM Attack

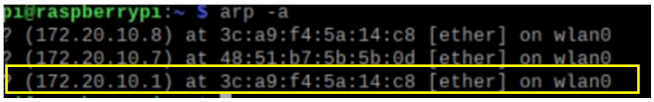

Fig. 15. MAC Address table in the victim's device after the MiTM Attack

Fig. 16 shows the packets that have been successfully sniffed by the attacker using Bettercap. The attacker was able to obtain their private information, such as their username and password, because the victim was using an unsecured website. As a result, the attacker can exploit the information to gain access to the victim's login credentials. Unaware victims who use the same password for all their accounts risk experiencing loss and becoming vulnerable to cyberattacks.

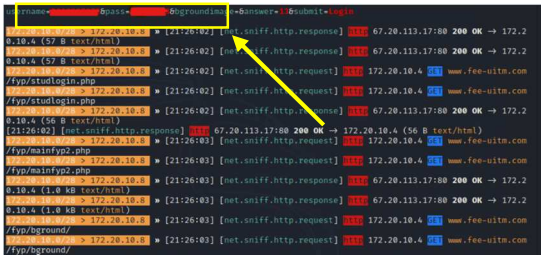

Fig. 16. The packets captured by the attacker through the MiTM attack using Bettercap

Since the attacker has successfully put themselves between the two communications, they can also cause disturbance to the victim. The attacker can use advanced MiTM tools such as Xerosploit to create more nifty attacks. In Fig. 17, when the victim surfs these websites, the page starts shaking uncontrollably and causes disturbance to the victim. The victims are unaware that they are under attack and might think that their monitor display or Internet is causing these issues.

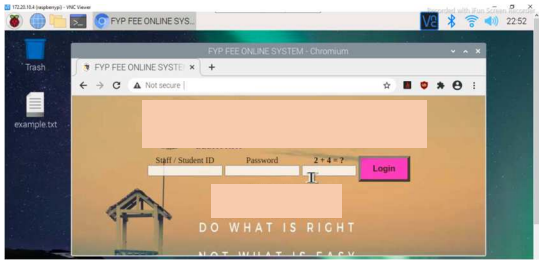

Fig. 17. The website was shaken due to a MiTM attack using Xerosploit

Another attack that can be done in Xerosploit is replacing images of the websites the victim visits. Before the attack, the victim can surf the web pages with no disturbance, as seen in Fig. 18. However, once the attack is executed, all of the images will be replaced with the image the attacker chose, as shown in Fig. 19. MiTM attacks can be avoided by taking a few measures, such as avoiding public Wi-Fi connections that are not password-protected, as an attacker may have configured them. In addition, avoid using unsafe websites; even if it is secured, users should log out when not in use. Users can also utilize a Virtual Private Network (VPN) to ensure secure communications and enable two-factor authentication. Emails are similarly vulnerable to attack. Consequently, utilizing SSL/TLS can help to keep them safe.

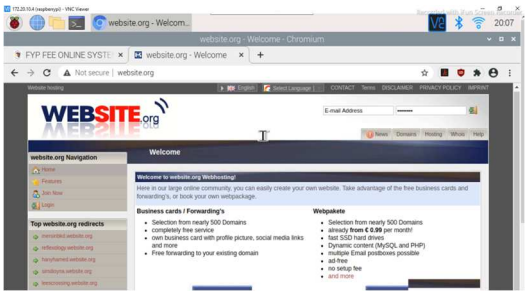

Fig. 18. The website before the MiTM attack using Xerosploit

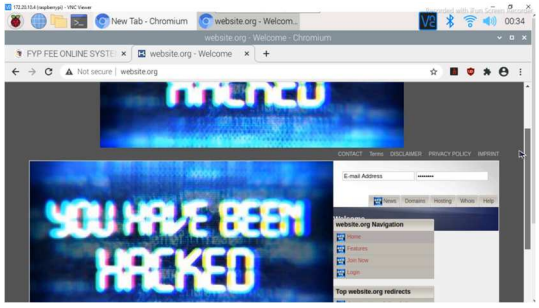

Fig. 19. The website after the MiTM attack using Xerosploit

IV. Conclusion

In conclusion, this project has successfully met the objectives specified. Penetration testing on the Raspberry Pi reveals many vulnerabilities in most embedded or IoT-related controllers such as drones, smart locks, smart bulbs or many other IoT devices. This highlights the demand for embedded system security aspects in all IoT-related applications. Mitigations to the attacks were also discussed to improve the security level in IoT devices. In addition, society may be at peace knowing that their personal information is safeguarded and not exposed to someone that can bring them trouble.

For future work, it is recommended to extend the diversity of this project by focusing on an IoT-based smart home network that could help identify the vulnerabilities and threats

in the network. Other than that, energy profiling could also be done in the future to detect the energy consumption when the IoT devices are under cyber-attacks.

ACKNOWLEDGMENT

The authors would like to thank the College of Engineering, Universiti Teknologi MARA, 40450 Shah Alam, Selangor for the funding in publishing this paper.

REFERENCES

[1] S. Siboni et al., "Security Testbed for Internet-of-Things Devices," in *IEEE Transactions on Reliability*, vol. 68, no. 1, March 2019, pp. 23-44, doi: 10.1109/TR.2018.2864536.

[2] F. Meneghello, M. Calore, D. Zucchetto, M. Polese and A. Zanella, "IoT: Internet of Threats? A Survey of Practical Security Vulnerabilities in Real IoT Devices," in *IEEE Internet of Things Journal*, vol. 6, no. 5, Oct. 2019, pp. 8182-8201, doi: 10.1109/JIOT.2019.2935189.

[3] S. Shiaeles, N. Kolokotronis and E. Bellini, "IoT Vulnerability Data Crawling and Analysis," in *2019 IEEE World Congress on Services (SERVICES)*, 2019, pp. 78-83, doi: 10.1109/SERVICES.2019.00028.

[4] J. Zaddach, L. Bruno, A. Francillon, and D. Balzarotti, "Avatar: A Framework to Support Dynamic Security Analysis of Embedded Systems' Firmwares", in *Network and Distributed System Security Symposium* (*NDSS)*, vol. 14, 2014, pp. 1-16, doi: 10.14722/ndss.2014.23229

[5] D. Davidson, B. Moench, T. Ristenpart, and S. Jha, "On Firmware: Finding Vulnerabilities in Embedded Systems Using Symbolic Execution," in *22nd USENIX Security Symposium (USENIX Security 13)*, 2013, pp. 463-478.

[6] J. Liu, Y. Xiao and C. L. P. Chen, "Authentication and Access Control in the Internet of Things," *2012 32nd International Conference on Distributed Computing Systems Workshops*, 2012, pp. 588-592, doi: 10.1109/ICDCSW.2012.23.

[7] W. Shang, Q. Ding, A. Marianantoni, J. Burke and L. Zhang, "Securing Building Management Systems Using Named Data Networking," in *IEEE Network*, vol. 28, no. 3, May-June 2014, pp. 50-56, doi: 10.1109/MNET.2014.6843232.

[8] A. Costin, "Security of CCTV and Video Surveillance Systems: Threats, Vulnerabilities, Attacks, and Mitigations," in *Proceedings of the 6th International Workshop on Trustworthy Embedded Devices*, 2016, pp. 45-54, doi: 10.1145/2995289.2995290

[9] J.-b. Hou, T. Li, and C. Chang, "Research for Vulnerability Detection of Embedded System Firmware," *Procedia Computer Science,* vol. 107, 2017, pp. 814-818, doi: 10.1016/j.procs.2017.03.181.

[10] E. Al Neyadi, S. Al Shehhi, A. Al Shehhi, N. Al Hashimi, M. Qbea'H and S. Alrabaee, "Discovering Public Wi-Fi Vulnerabilities Using Raspberry pi and Kali Linux," *2020 12th Annual Undergraduate Research Conference on Applied Computing (URC)*, 2020, pp. 1-4, doi: 10.1109/URC49805.2020.9099187.

[11] L. Liang, K. Zheng, Q. Sheng and X. Huang, "A Denial of Service Attack Method for an IoT System," *2016 8th International Conference on Information Technology in Medicine and Education (ITME)*, 2016, pp. 360-364, doi: 10.1109/ITME.2016.0087.

[12] Q. Chen, H. Chen, Y. Cai, Y. Zhang and X. Huang, "Denial of Service Attack on IoT System," 2018 9th International Conference on Information Technology in Medicine and Education (ITME), 2018, pp. 755-758, doi: 10.1109/ITME.2018.00171.