Assessing and Mitigating Network Vulnerabilities in Philips Hue and Nest Protect Smart Home Devices

Arvind Sredhar[1], Adil Khan[2], Abdul Rehman Gilal[3], Aeshah Alsughayyir[4], Abdullah Alshanqiti[5], Bandeh Ali Talpur[6]
School of Digital, Technologies and Arts, Staffordshire University, United Kingdom[1]
IBISC, Université Paris-Saclay, 91020 Evry, France[2]
Department of Computer Science, Sukkur IBA University, Sukkur 65200, Sindh, Pakistan[2]
Knight Foundation School of Computing and Information Sciences, Florida International University, United States[3]
College of Computer Science and Engineering, Taibah University, Madinah, Kingdom of Saudi Arabia[4]
Faculty of Computer and Information Systems, Islamic University of Madinah, Kingdom of Saudi Arabia[5]
School of Computer Science and Statistics, Trinity College, Dublin, Ireland[6]

***Abstract*—The Internet of Things (IoT) has gained momentum across various sectors, particularly in the consumer market with the adoption of smart devices. IoT extends internet connectivity to physical devices, enabling control via smartphones, environmental sensing, and updates. However, smart home devices are susceptible to cyberattacks due to vulnerabilities, lack of monitoring, and built-in security. They can also participate in botnets, leading to large-scale attacks. Vulnerabilities in these devices may exist at the sensing, network, or application layers, impacting data confidentiality, integrity, and service availability. This research aims to identify network-layer vulnerabilities affecting the 'Availability' of Philips Hue and Nest Protect. By establishing a test environment, the baseline behavior of these devices is examined, followed by scans for open ports and services to detect network-based threats. Volumetric flood attacks are then conducted to assess susceptibility, and findings are shared to define the devices' default security posture. The research also addresses security issues related to home routers and aims to reduce the attack surface of smart home devices through isolation and network-level protection. This involves deploying a Firewall to isolate smart devices from non-IoT devices and prevent intrusions.**

Keywords*—*Internet of Things (IoT); Smart Home Devices (SHDs); network vulnerability assessment; Philips Hue; Nest Protect

I. INTRODUCTION

The Internet of Things (IoT) has been an important technological revolution that has enabled the emerging Industry 4.0 [1]. By extending the capabilities of physical devices used in our daily life to the Internet, IoT, with the use of standard Internet Protocols, has allowed communication between humans and things [2], [3]. Smart devices are context-aware, perform autonomous computing, and connect using wired or wireless mediums [4]. Since these devices carry out only a limited set of tasks, they have low hardware specifications [5]. Some smart devices require intermediate nodes to communicate on the internet. Owing to these reservations, smart devices use lightweight communication methods and lack complex built-in security schemes [6].

In a smart home, internet connectivity is extended to consumer appliances [7] such that human–device communication can occur seamlessly. For example, the Philips Hue smart lights can be controlled using a smartphone, and the Nest Protect alerts the user whenever smoke is detected [8], [9]. Smart Home Devices (SHDs) have gained immense popularity and have become an important part of modern living. In 2019, an estimated 8.6 billion devices were connected worldwide [10] and this number is expected to increase to 29.4 billion by 2030. The worldwide unit shipments of SHDs for 2021 were nearly 900 million [11], [12], [13]. While one set of statistics shows how popular SHDs are becoming, another set of statistics questions how secure these devices are. In 2022, the worldwide annual number of IoT-related cyberattacks amounted to 112 million [14].

Security is a vital requirement for all communication systems. Numerous security solutions that have evolved have focussed only on traditional computing. SHDs are the least secure of internet hosts [15] and both their widespread growth and heterogeneity have opened new and significant attack surfaces [16], [17]. Hence, the security of IoT devices is more critical than that of traditional computing devices [2], [18]. As in [19] cautioned, a compromise in SHD security not only affects the digital world but can also have serious implications in the physical world, causing harm to people.

In an enterprise, the complex task of monitoring and managing both IoT and non-IoT devices is handled by automated solutions and dedicated technical staff. In contrast, the security responsibility of a smart home falls on the user. Unfortunately, many consumers lack awareness about the potential risks these connected devices can cause and fail to implement adequate security measures [20]. Furthermore, many manufacturers of consumer network devices may not find an incentive to release frequent updates and patch vulnerabilities [21]. These gaps form the primary motivation to conduct this research and find out whether the devices we use have vulnerabilities. Hence, this research aims to secure two SHDs – the Philips Hue Smart Lighting and the Nest Protect Alarm.

The commonly prevalent security issues are highlighted by foundations such as the OWASP [22] and recommendations are provided by institutions such as the UK Government, ENISA, and the ETSI [23], [24], [25]. However, owing to inexperience in Cybersecurity and lack of security-focussed

development, manufacturers still flood the market with insecure devices. For example, the DDoS attack against Dyn, in 2016, was conducted using compromised consumer IoT devices. This suggests that vulnerabilities in a SHD, when not addressed, can turn it into digital weapon [26].

Davis et al. [27] categorize SHD vulnerabilities as Physical, Network, Software, and Encryption. While software and encryption related vulnerabilities are more manufacturer-centric, this research focuses on the network-level vulnerabilities. It is a cause for great concern that SHDs lack security standards and that vulnerabilities get exposed only during usage. It is imperative to perform vulnerability assessments of SHDs and identify network insecurities. Two studies have assessed the security posture of both the Philips Hue and the Nest Protect and have provided security ratings [20], [28]. Both Copos et al. [29] and Yadav et al. [30] have conducted a traffic analysis of the Nest Protect. The first objective of this research is to setup a test network to assess the network insecurities in the Philips Hue and the Nest Protect. Based on the vulnerabilities identified, attacks are launched against the devices and the responses are recorded. This forms the second objective of this research.

A typical Smart Home follows a flat network architecture – both IoT and non-IoT devices are on the same subnet served by a home Gateway. It is highly possible that such a coexistence may open new avenues for cyberattacks [31]. In addition, the home Gateway is the most compromised device and its services may increase the attack surface [15]. Hence, this research also discusses vulnerabilities in the home Gateway and how those could increase the risk factor.

Generally, devices in a traditional network are secured using three approaches namely: 1) device-level protection, 2) isolation, and 3) network level protection. Unlike computers, SHDs lack power and computing resources to apply device-level protection [31], [32]. Considering the lack of first-line defence mechanisms, the third objective of this research is to apply a second line defence mechanism. The research proposes that by isolating the SHDs and applying network-level protection, the attack surface can be reduced. Studies suggest that Firewall, IDS, and IPS as solutions to the threats occurring at the network layer [33], [34]. A security solution that both acts as a Firewall and that has the capability of an IPS is deployed. The objective is to segregate devices into separate zones and by apply access rules such communications among the IoT and non-IoT devices are curtailed. Tests are performed to validate if the artefact could successfully reduce the attack surfaces of the SHDs.

II. Background Information

The OWASP Project provides an overview of the top 10 security issues [22] found in IoT. This can be taken as a guideline to assess the type of insecurities found in SHDs. Especially, insecure network services that is ranked as a serious security issue pertains to the unneeded or insecure services in the device. Such a vulnerability can impact Confidentiality, Integrity, and Availability.

itaThreats to SHDs need not always arise from the internet; As proved by Chan et al. [35] a threat actor who has access to the internal network can misuse a vulnerability for larger attacks. As pointed out by Loi et al. [16], lack of vigour in fixing vulnerabilities in devices, lack of awareness among consumers about potential risks, and lack of network isolation or separate security solutions in home networks are seen as incentives by threat actors. With access to LAN, malicious actors may not only fingerprint every device using tools but also launch passive or active attacks.

The first aspect of this research is focussed on analysing the network-level vulnerabilities in Philips Hue Smart Light and Nest Protect Smoke Alarm. The work of Loi et al. [16] informs that both the Hue Light and the Nest Protect have open TCP and UDP ports and that the Hue Light is more vulnerable. In the case of Philips Hue, the authors indicate that the open TCP port 80 is the vulnerable port. This claim is further supported by CVE-2018-7580 [36] that a SYN-Flood DoS can result in an consume resources of the Philips Hue in an uncontrolled manner, resulting in unavailability of service. The SYN-Flood which is a Protocol based attack exploits the TCP 3-way Handshake by flooding the endpoint with excessive SYN packets. When the OS exceeds the threshold of concurrent connections it can maintain, it denies access to TCP services [37].

Network tools such as 'hping3' aid in generating large number of packets against a target [38], and when proper security measures are not employed, such floods result in unavailability of service – in this case a user may not be able to switch on/off the smart lights. Although the vulnerable TCP port 80 was reported for SYN-Flood, what other forms of attacks or information can be gathered from this open port is question that needs to be addressed. TCP port 80 falls under the well-known ports category [39], denoting that a server providing http service is listening on this port. Hence, this vulnerability not only allows a threat actor to conduct a Protocol DoS but also an Application Layer DoS against the device.

In the case of Nest Protect Smoke alarm, Loi et al. [16] state that numerous UDP ports in the 'registered ports' category remain open. Their work does not provide specific information about these ports and the functions. This research includes finding more information about those open ports and other direct flood attacks that impact the Nest Protect.

Although identifying open ports in each SHD provides information about the associated protocol and service, this process must be augmented with the capture the network traffic. This is the second aspect of this research. Capturing network traffic with packet sniffing tools such as Wireshark provides more insights. By studying the ingress and egress traffic of the devices, one can chart out not only the device, domains, and services contacted but also the frequency of such conversations. This research involves capturing traffic for both the Philips Hue and the Nest Protect and presents the baseline behaviour of these devices.

The third aspect of this research discusses about the impact of SHD vulnerabilities on a network, in general. As mentioned in the first paragraph, typically, home networks are 'flat networks' without any segmentation or isolation. This model in which traditional computers coexist with SHDs only increases

security concern. Without isolation a compromised device can inflict damage on other IoT and non-IoT devices. Hence, this research proposes a solution that applies not only for the Philips Hue and the Nest Protect, but also for SHDs in general. By identifying vulnerabilities, understanding network traffic patterns, and by isolating IoT devices and applying a firewall the attack surface can reduced.

III. Examine, Attack, and Implementation

This research aims to identify network-layer vulnerabilities affecting the 'Availability' of Philips Hue and Nest Protect. The objective of this study is to evaluate the security of the Philips Hue smart light system and the Nest Protect smoke detector alarm as network-connected devices. The research begins by examining these devices using open-source tools to identify vulnerabilities and understand how they impact the devices' services and other connected devices. By using Philips Hue and Nest Protect as case studies, this study seeks to provide insights into the broader challenges of securing smart home devices in networked environments.

The Philips Hue Bridge version 2.1 functions on Mains power supply, connects to the network using Ethernet, and communicates with bulb using Zigbee protocol [36]. The Nest Protect is a second-generation smoke alarm that is powered by batteries, connects to the network using the IEEE 802.11 b/g/n 2.4GHz Wi-Fi standard, communicates with smartphones using BLE, and exchanges information with other connected Nest products using the IEEE 802.15.4 2.4 GHz standard [8].

The heterogeneity of IoT devices is quite evident that the SHDs used in this research vary in terms of power and communication technologies. Such heterogeneity, consequently, has a bearing on the lab network setup used for assessing the SHDs. For security reasons, the lab setup uses dedicated desktop, laptop, and networking devices.

A. Examining the Philips Hue Smart Lighting

As shown in Fig. 1, the TP-Link Archer C60 wireless router acts as a gateway and leases IP addresses. The TP-Link TL-SG108E switch has built in functionality for port mirroring and is used for traffic capture. Kali Linux 2023.2 which has the tools to examine the Hue Bridge is installed on the Raspberry Pi 4B. The Hue App to control the smart bulbs is installed on an iPhone 6s running iOS 15.7.x. Devices that require authentication are configured with a 12-character password that includes uppercase and lowercase characters and numbers.

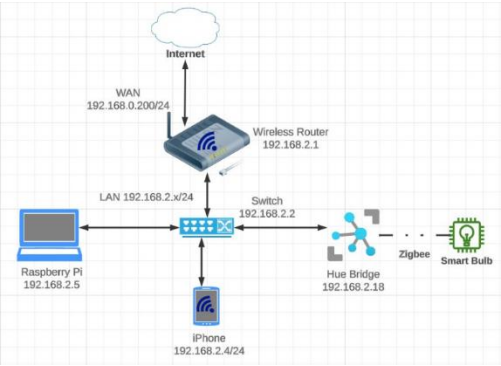

Fig. 1. Network setup – hue port scan and packet capture.

From the Raspberry Pi, a subnet scan was conducted. The network ports that are open in each of the five devices mentioned above are listed. TCP ports 80, 443, and 8080 are open in the Hue Bridge. Using the Nmap scan options -sS and -sT, a TCP SYN and TCP Connect scans are conducted. Using the -p switch, ports 1-65535 were scanned. Results show the same set of open TCP ports as the subnet scan. Further investigation carried out using an Nmap command with switches -sC -sV -O against the Hue reveals the 'Service' listening on the open TCP ports. As opposed to the SYN and Connect scan, the Fingerprint scan includes only the 1000 most popular ports. A web server is listening to TCP port 80, 443, and 8080. Out of these three open ports, 80 and 8080 use the plaintext HTTP whereas 443 uses SSL – a protocol that uses encrypted link between the client and the web server [37].

All the above scans have listed information only about the TCP ports. However, UDP based open may also be employed in this device. To find the open UDP ports we conduct a UDP scan using Nmap with the -sU switch. UDP scans can be very slow, scanning 65535 ports took around 18 hours. The UDP scan reveals three more open ports (1900, 5353, and 5540).

B. Examining Network Behaviour of the Philips Hue Smart Lighting

Taking advantage of port mirroring, ingress and egress traffic of the Hue Bridge was captured using both TCPDump and Wireshark. Traffic was captured based on 4 scenarios – 1. Powering on the Hue Bridge, 2. Idle Operation for 60 minutes, 3. Switching ON/OFF the bulbs using Wi-Fi, and 4. Switching ON/OFF the bulbs using 4G mobile data.

As the Hue Bridge begins to operate, it contacts the domains, almost all the domains/services are Cloud-based services hosted by AWS, Google Cloud, and Alibaba Cloud. The packet capture was repeated on different days, and it was observed that although the domains contacted were the same, the Public IP Addresses of those providers did not remain a constant. Although this observation did not hold for all the services hosted on Google Cloud, it holds true for services hosted on AWS.

HTTP traffic was found only at two instances 1) During the initial pairing between the Bridge and the smartphone and 2) At regular intervals between the Bridge and the domain www.ecdinterface.philips.com. Investigating the HTTP traffic between the Hue Bridge and the Cloud-service, reveals the type of device, its MAC address, and Public Key details. This is seen as a vulnerability as this can be of value to a malicious actor eavesdropping on the network.

C. Vulnerabilities of the Philips Hue Smart Lighting

As discussed above, the Philips Hue has TCP ports 80, 443, 8080 and UDP ports 1900, 5353, and 5540 open. The number of open ports may signify more vulnerabilities, resulting in an increased the attack surface. Loi *et al*. [16] in their research have stated that TCP port 80 in the Hue is vulnerable, and from CVE-2018-7580 [36] it is evident that a SYN-Flood against port 80 render the Hue unresponsive.

The SYN Flood is a technique that misuses the TCP 3-way handshake by sending large amounts of SYN packets to an

endpoint. The device responds to each SYN packet and keeps waiting with open connections expecting a graceful connection closure. On the contrary, the closure may not arrive, resulting in exhaustion of resources and denial of new connections.

We have also validated this technique by creating test environment that includes three virtual computers running Kali Linux, Ethernet Switch that supports Port Mirroring, Hue Bridge, and the smartphone with the Hue App (see Fig. 2).

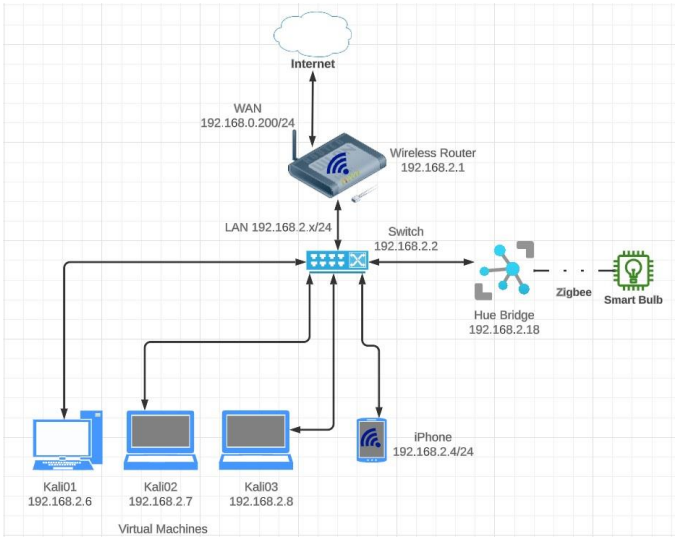

Fig. 2. Lab setup for DoS attack.

It is interesting to note that the first two sets of SYN-Floods which had a count of 500 and 750 from each machine did not have any effect. Despite the excessive amount of traffic, the Bridge and Hue App continued to remain functional throughout the test. However, in the case of the third test in which the count was raised to 1000, the impact rendered the service unavailable.

Two sets of SYN-Flood from each virtual machine was launched against TCP port 8080. However, there was no impact on the 'Availability' of the service. This proves that, with respect to SYN-Floods, TCP port 8080 is indeed the vulnerable port (Loi et al., [16]). Additionally, Hue can be subjected to HTTP and ICMP attacks. 'SlowHTTPTest' is a tool that simulates Application Layer DoS attacks and is part of Kali Linux. Using this tool two DoS attacks with 200 and 500 connections were launched, both tests disrupted the availability of the service (see Fig. 3).

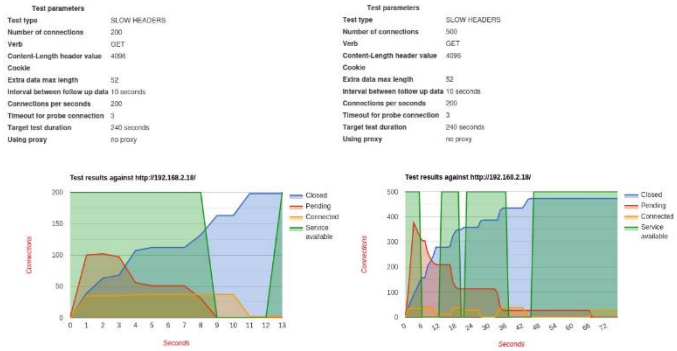

Fig. 3. DoS attack results.

An important observation that has not been mentioned in research articles is that the Hue Bridge suffers from ICMP Flood. Three sets of ICMP Flood tests were conducted from the virtual machines and the service was unavailable during each test. Loi et al. [16] state that the Hue Bridge remains protected from ICMP DoS. However, the test result shows that the impact of ICMP flood is worse compared to SYN and HTTP Floods.

D. Examining Nest Protect Smoke Alarm

In contrast to the Hue Bridge, the Nest Protect uses Wi-Fi and not Ethernet. Hence to address this the lab setup was changed to conduct the Port Scans and Traffic Analysis. As shown in Fig. 4, the Raspberry Pi was used as a Wi-Fi Access Point to which the Alarm and the smartphone connect. With this setup, traffic passing through the WLAN adapter can be captured.

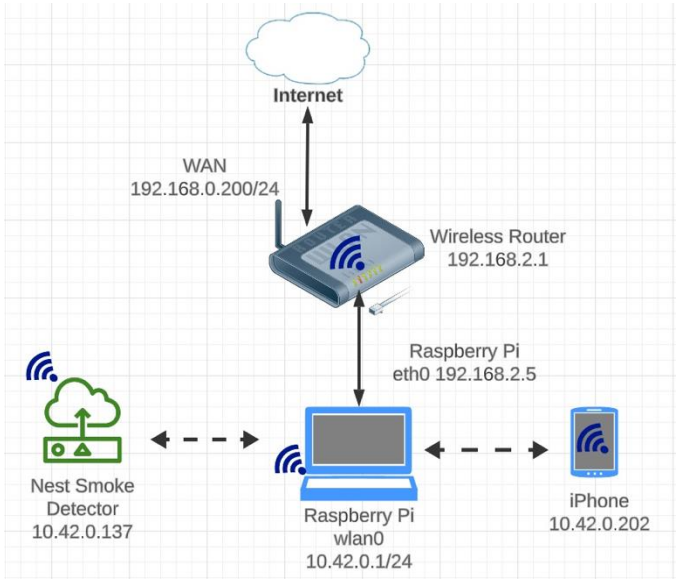

Fig. 4. Lab setup for Nest Protect.

During the initial phases of testing, it was observed that TCP SYN, Connect and Fingerprint scans did not yield any result. It was assumed that all the ports in the Nest Protect Alarm were either filtered or closed. A Ping scan of the subnet would result in displaying two hosts - the Raspberry Pi and the smartphone, but not the Alarm. It was assumed that the manufacturer had also locked ICMP Request/ Reply. Later, during a second round of analysis, it was found that the Nest Protect remains awake only for a duration of 120 seconds and goes to sleep-mode. By pressing the button on the smoke detector, the device is again activated, and communication is restored. During sleep-mode, network communication is cut off, possibly, as a power saving measure. TCP scans were set to defaults and performed within 120 seconds of each activation. It can be noted that the IP addresses of the devices during the initial phase and the second phase are different since the test environment was set up again. In contrast to the results present by Loi et al. [16], the targeted UDP port scans were not able to find any open ports in the 17000-20000 port range. It is assumed that the manufacturer must have closed these ports in a software update.

E. Examining Network Behaviour of Nest Protect Smoke Alarm

Unlike Philips Hue, both the Nest smoke detector and the smartphone app contact Cloud services. Certain surprising observations from traffic analysis reveal that the Nest smoke detector generates comparatively very less traffic – also validated by Yadav *et al.* [30]. This condition is true even when a safety check is initiated. The only HTTP traffic that was observed was between the alarm and clients.l.google.com. Rest of the traffic generated from the alarm were only TCP that used Dynamic port numbers at the source and Registered port 11095 at the Cloud servers. Further investigation reveals that Nest Protect uses the 'Weave' protocol to connect with its Cloud servers [40].

The smartphone with the Nest App makes an alarming amount of NTP requests to Google's Time Servers. However, the Nest App uses only TCP and TLS encrypted communication to all the servers it contacts. The communication with the alarm occurs only when it is active or manually invoked by pressing the button on the alarm or by using the App. The alarm does respond to ICMP until the time it is active.

From this section, it can be inferred that by comparison the Philips Hue has more open ports than the Nest Protect. This translates to more vulnerabilities and increased attack surface. In the next section, DoS attacks will be launched against the Philips Hue and the Nest Protect Alarm, and the impact of those attacks will be recorded. Solutions to reduce the attack surface and mitigate the attacks are implemented and their efficiency is validated.

F. Vulnerabilities of the Nest Protect Smoke Alarm

Compared to the Hue Bridge, it can be claimed that the Nest Protect Alarm has a lower attack surface. Since the Nest Protect does not reveal ports and services in use, it is difficult for a threat actor to attack. However, the Alarm is still vulnerable to ICMP Flood. During an ICMP Flood the Nest App could not establish a connection with the device. However, the device remains active only for a duration of 120 seconds and the chances of successful ICMP Flood attacks remain slim. As stated by Notra et al. [41] the Nest Protect Alarm is indeed a secure product. It is possible that the Nest Protect could be vulnerable to sleep-depravation attacks, but those attacks are beyond the scope of this research.

G. Implementing Solutions to Secure SHDs

Securing a device is a two-fold process – host-based security and network-based security. SHDs lack built-in security owing to their size, computational power, and power consumption. This means that unlike computers security cannot be enhanced using host-based security solutions. Also, any shortfall in security, such as open ports or unnecessary services cannot be fixed (Shirali-Shahreza and Ganjali, [42]). Since device-level security cannot be applied, the artefact applies the other two security approaches, 1) isolation and 2) network-level protection, as suggested by Hamza et al. (Hamza, Gharakheili and Sivaraman, [31]).

1) Isolation and network-level protection of SHDs: As the research involves real SHDs, to implement a solution a device had to fulfil certain conditions: it must be affordable, portable, must serve the Ethernet-based Philips Hue and the Wi-Fi based Nest Protect, and must fit in the existing network without major changes. Based on these conditions a Raspberry Pi 4B was chosen. To avoid license costs, a Linux Kernel based distribution was the preferred choice of OS for the Raspberry Pi. Open-source tools such as Snort, pfSense, OpenWrt, and IPFire were considered. pfSense was not tested as it supported only certain architectures. Ubuntu Desktop 22.04 LTS was installed on the Raspberry Pi as installing Snort is a straight-forward process in Ubuntu. However, owing to the sluggish behaviour of the OS and difficulties in successfully installing configuring dependent packages such as 'dnsmasq' and 'hostapd' the idea was aborted. Since OpenWrt and IPFire both were supported, both tools were installed in separate SDXC cards and tested.

Both OpenWrt (2023) and IPFire (2023) are open-source and community-supported products that ensure security by default. Both the products are compatible with the Rapberry Pi, satisfy conditions for isolation and network-level protection, and offer GUI. However, IPFire was chosen over OpenWrt as the former is designed and optimised to be a Firewall whereas the latter includes Firewall functionality. Unlike commodity Home Gateways and OpenWrt, IPFire enables HTTPs based GUI by default.

It is straightforward to choose a network configuration in IPFire that supports WAN, LAN, and WLAN interfaces required for this research. The WAN interface or the RED zone has been connected to the existing network and the SHDs are isolated from each other and from the existing home network. The Hue Bridge is part of the GREEN zone (Subnet-02 - 10.100.100.0/24) and the Nest Protect is part of the BLUE zone (Subnet-03 – 10.100.200.0/24) as shown in Fig. 5. Although the installation of IPFire in Raspberry Pi is not straightforward, it can still be achieved with support from the developer [43], a process that is easier in OpenWrt.

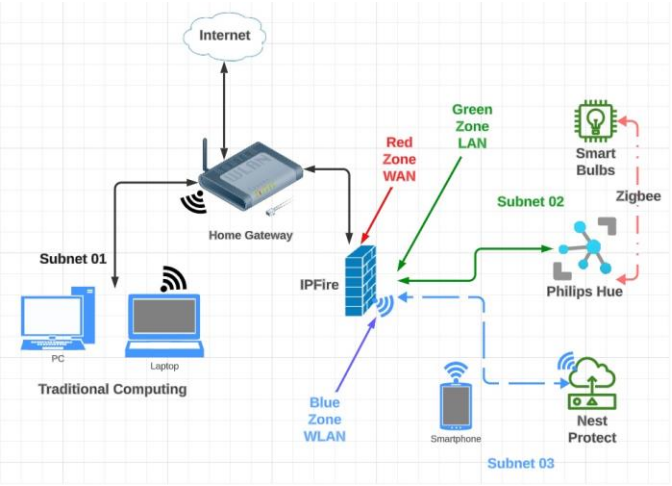

Fig. 5. Network setup and artefact placement.

Through the 'PakFire' module additional packages such as 'hostapd' which enables the Wi-Fi access point can be installed. By default, IPFire enables MAC filtering for wireless

clients. Hence every new device connecting to Wi-Fi must be approved. This security features disallows rogue devices connecting to Wi-Fi. Although IPFire can be remotely accessed via SSH, as a security feature it is disabled by default.

IPFire's SPI firewall, based on 'netfilter', by default restricts traffic between the zones [44]. In addition, the Firewall provides 'IP Address Blocklists' to deal with traffic based on the reputation of IP Address [45], and 'Location Block' to block incoming connection from certain Geo-locations [46].

One of the most important features of IPFire is its ability to function as an IPS. IPFire employs Suricata [47], an open-source software for intrusion prevention [48]. Although, this feature is disabled by default, it can be enabled on more than one interfaces. Traffic passes through the IPS before it is sent to the Firewall, and malicious traffic is dropped by the IPS. IPS works based on 'Rulesets' and IPFire allows an user to choose more than one ruleset. Some community rulesets are free whereas some need add a subscription. By this, IPFire achieves the functionality of Snort IPS. The Firewall, IP Blocklists, and IPS have separate logs that can be accessed through the GUI. Connections are tracked and are displayed in the GUI. In addition, CPU load, Memory usage and Processes are displayed graphically [49].

IV. Conclusion And Future Work

Smart Home systems are becoming more popular and the rate of adoption of these devices has been tremendous. SHDs make our life easier by allowing physical devices to be controlled over the internet. However, SHDs may have vulnerabilities and can introduce new challenges to home network security. The DDoS attack against Dyn that was conducted using compromised consumer IoT devices is proof that smart devices can participate in larger attacks.

Typically, IoT and non-IoT devices coexist in home networks. In many cases SHDs remain unmonitored and consumers are unaware of the security issues that may exist in these devices. Weak authentication methods and insecure network services have been the top security issues found in IoT devices. Hence, it is imperative to study the insecurities in SHDs and deploy security solutions to reduce the attack surface.

Vulnerabilities may reside in any of the three layers in the IoT architecture. However, this research focussed in identifying the vulnerabilities at the network layer. A test lab was setup to examine the Philips Hue and the Nest Protect. The baseline behaviour of each of these devices were recorded. It is evident that both the devices depend on various Cloud-hosted services. Although both the devices use secure protocols and encrypt application data, the Philips Hue still uses HTTP based communication to a cloud service.

The research then involved Nmap port scans which revealed the open ports and the associated services. Abusing these ports and services by sending excessive amounts of traffic can directly impact the 'Availability'. This was demonstrated by conducting volumetric flood attacks against the Philips Hue. Owing to open ports, a threat actor can launch SYN Flood, HTTP Flood, and ICMP Flood against the Philips Hue, turning the device unresponsive.

On the other hand, services such as UPnP that advertise the capabilities of a device on the network can be misused, exposing devices to the internet. In the case of the Philips Hue, the UPnP service exposes the device's unique identifiers. Such details can result in spoofing attacks. It is evident that the Nest Protect alarm is, comparatively, a safer device as it does not expose its ports and services. Still, an ICMP Flood launched against the Nest Protect turned the device, temporarily, unresponsive. It is possible that the Nest Protect can suffer from sleep deprivation attacks, however such attacks were outside the scope of this research.

Unlike computers, IoT devices cannot be protected using device-based security solutions. Hence, the focus was shifted to protecting these devices through isolation and applying network-level protection. In addition, the Philips Hue and the Nest Protect are heterogeneous – the former uses Ethernet and the later Wi-Fi for communication. Hence the solution must encompass both standards. Deploying IPFire, a powerful open-source Netfilter based SPI Firewall on a portable device like Raspberry Pi 4B fulfils the requirements. IPFire separates the flat home network into three zones, and by default curtails communications between the LAN and WAN zones. Through Firewall rules, it was demonstrated that complete isolation between LAN and WLAN zones can be achieved.

Unlike commodity Gateways and Wi-Fi Routers, IPFire neither has in-built UPnP nor does it allow the service to be installed. This ensures that malicious programs cannot open ports without user consent. The Suricata IPS engine is one of the important aspects of IPFire since packets are analysed by the IPS and any malicious traffic is dropped before reaching the firewall. This feature ensures that only legitimate traffic reaches the SHDs. In addition, IP Blocklists can also be configured to drop traffic from IP Addresses with poor reputation. IPFire by default enables DNSSEC and the user can enable DNS over TLS. Such features reduce the chances of DNS spoofing and cache poisoning. With these features, IPFire reduces the attack surface of the Philips Hue and the Nest Protect. Although deploying a Firewall and IPS ensure security, improving the efficiency of such systems is a continuous process that involves scrutinising the Firewall and IPS logs and applying relevant rulesets.

Insecurities exist in all layers of the IoT architecture, but this research was limited only to the network layer and to certain types of DoS attacks. In the future work, sensing layer related vulnerabilities will also be included. Another limitation is, IPFire supports only IP address-based access rules. A Firewall that supports Fully Qualified Domain Name (FQDN) based rules will involve less overhead since Cloud providers tend to assign various public IP addresses for hosted services. The methods followed in this research can be expanded to include other devices. Hence, the future work will include devices such as the Smart TV.

References

[1] Gazis, 'What is IoT? The Internet of Things explained', ACADEMIA Letters, vol. 1003, pp. 1–8, Jun. 2021, doi: 10.20935/AL1003.

[2] O. Garcia-Morchon, S. Kumar, and M. Sethi, 'Internet of Things (IoT) Security: State of the Art and Challenges', Internet Engineering Task Force, Request for Comments RFC 8576, Apr. 2019. doi: 10.17487/RFC8576.

[3] L. Fetahu, A. Maraj, and A. Havolli, 'Internet of Things (IoT) benefits, future perspective, and implementation challenges', in 2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO), May 2022, pp. 399–404. doi: 10.23919/MIPRO55190.2022.9803487.

[4] M. Silverio-Fernández, S. Renukappa, and S. Suresh, 'What is a smart device? - a conceptualisation within the paradigm of the internet of things', Vis. in Eng., vol. 6, no. 1, p. 3, May 2018, doi: 10.1186/s40327-018-0063-8.

[5] M. G. Samaila, M. Neto, D. A. B. Fernandes, M. M. Freire, and P. R. M. Inácio, 'Challenges of securing Internet of Things devices: A survey', SECURITY AND PRIVACY, vol. 1, no. 2, p. e20, 2018, doi: 10.1002/spy2.20.

[6] M. Abomhara and G. M. Køien, 'Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders and Attacks', Journal of Cyber Security and Mobility, pp. 65–88, May 2015, doi: 10.13052/jcsm2245-1439.414.

[7] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, 'Internet of Things (IoT): A vision, architectural elements, and future directions', Future Generation Computer Systems, vol. 29, no. 7, pp. 1645–1660, Sep. 2013, doi: 10.1016/j.future.2013.01.010.

[8] Google, 'Nest Protect 2nd generation technical specifications - Google Nest Help', Nest Protect 2nd generation technical specifications. Accessed: Aug. 30, 2023. [Online]. Available: https://support.google.com/googlenest/answer/9229922?hl=en-GB&ref_topic=9361988&sjid=10364023325044099712-EU#

[9] Philips, 'How Smart Lighting works', Philips Hue US. Accessed: Aug. 27, 2023. [Online]. Available: https://www.philips-hue.com/en-us/explore-hue/how-it-works

[10] Transforma Insights, 'IoT connected devices worldwide 2019-2030', Statista. Accessed: May 17, 2023. [Online]. Available: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/

[11] IDC, 'Global smart home device shipments 2018-2026', Statista. Accessed: May 17, 2023. [Online]. Available: https://www.statista.com/statistics/920679/smart-home-device-shipments-worldwide-by-category/

[12] A. A. Khan, A. A. Wagan, A. A. Laghari, A. R. Gilal, I. A. Aziz, and B. A. Talpur, 'BIoMT: A state-of-the-art consortium serverless network architecture for healthcare system using blockchain smart contracts', IEEE Access, vol. 10, pp. 78887–78898, 2022.

[13] A. Al-Ashmori et al., 'Classifications of sustainable factors in Blockchain adoption: a literature review and bibliometric analysis', Sustainability, vol. 14, no. 9, p. 5176, 2022.

[14] SonicWall, 'Annual number of IoT attacks global 2022', Statista. Accessed: May 17, 2023. [Online]. Available: https://www.statista.com/statistics/1377569/worldwide-annual-internet-of-things-attacks/

[15] J. Melzer, J. Latour, M. Richardson, A. Ali, and W. Almuhtadi, 'Network Approaches to Improving Consumer IoT Security', in 2020 IEEE International Conference on Consumer Electronics (ICCE), Jan. 2020, pp. 1–6. doi: 10.1109/ICCE46568.2020.9043121.

[16] C. Bellman and P. C. van Oorschot, 'Analysis, Implications, and Challenges of an Evolving Consumer IoT Security Landscape', in 2019 17th International Conference on Privacy, Security and Trust (PST), Aug. 2019, pp. 1–7. doi: 10.1109/PST47121.2019.8949058.

[17] H. A. Ali, K. Shaikh, M. Chohan, K. F. Memon, M. Saleem, and A. Khan, 'Does Selection of Open Source Cloud Computing Platforms is a Confusing Task?', Accessed: Feb. 22, 2024. [Online]. Available: https://www.researchgate.net/profile/Hafiz-Ali-17/publication/340983379_Does_Selection_of_Open_Source_Cloud_Computing_Platforms_is_a_Confusing_Task/links/5ea87b2b92851cb26760c32c/Does-Selection-of-Open-Source-Cloud-Computing-Platforms-is-a-Confusing-Task.pdf

[18] A. R. Gilal, A. W. Adil Khan, M. Chohan, and H. A. Ali, 'Creating A Research Space In Software Engineering: Structure For Writing Introduction', International Journal of Scientific & Technology Research, vol. 9, p. 1373, 2020.

[19] E. Schiller, A. Aidoo, J. Fuhrer, J. Stahl, M. Ziörjen, and B. Stiller, 'Landscape of IoT security', Computer Science Review, vol. 44, p. 100467, May 2022, doi: 10.1016/j.cosrev.2022.100467.

[20] F. Loi, A. Sivanathan, H. H. Gharakheili, A. Radford, and V. Sivaraman, 'Systematically Evaluating Security and Privacy for Consumer IoT Devices', in Proceedings of the 2017 Workshop on Internet of Things Security and Privacy, in IoTS&P '17. New York, NY, USA: Association for Computing Machinery, 2017, pp. 1–6. doi: 10.1145/3139937.3139938.

[21] N. Nthala and I. Flechais, 'Rethinking home network security', European Workshop on Usable Security (EuroUSEC) 2018, Nov. 2018, doi: dx.doi.org/10.14722/eurousec.2018.23011.

[22] OWASP, 'OWASP Internet of Things | OWASP Foundation', OWASP Internet of Things (IoT) Top 10 2018. Accessed: May 17, 2023. [Online]. Available: https://owasp.org/www-project-internet-of-things/

[23] UK Government, 'Secure by Design Report', Mar. 2018. Accessed: Jul. 21, 2023. [Online]. Available: https://www.gov.uk/government/publications/secure-by-design-report

[24] ENISA, 'Good Practices for Security of IoT - Secure Software Development Lifecycle', ENISA. Accessed: Jul. 29, 2023. [Online]. Available: https://www.enisa.europa.eu/publications/good-practices-for-security-of-iot-1

[25] ETSI, 'Consumer IoT security', ETSI EN 303 645. Accessed: Jul. 29, 2023. [Online]. Available: https://www.etsi.org/technologies/consumer-iot-security?jjj=1690592385696

[26] S. M. Sajjad, M. Yousaf, H. Afzal, and M. R. Mufti, 'eMUD: Enhanced Manufacturer Usage Description for IoT Botnets Prevention on Home WiFi Routers', IEEE Access, vol. 8, pp. 164200–164213, 2020, doi: 10.1109/ACCESS.2020.3022272.

[27] B. D. Davis, J. C. Mason, and M. Anwar, 'Vulnerability Studies and Security Postures of IoT Devices: A Smart Home Case Study', IEEE Internet of Things Journal, vol. 7, no. 10, pp. 10102–10110, Oct. 2020, doi: 10.1109/JIOT.2020.2983983.

[28] A. Sivanathan, F. Loi, H. H. Gharakheili, and V. Sivaraman, 'Experimental evaluation of cybersecurity threats to the smart-home', in 2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Dec. 2017, pp. 1–6. doi: 10.1109/ANTS.2017.8384143.

[29] B. Copos, K. Levitt, M. Bishop, and J. Rowe, 'Is anybody home? inferring activity from smart home network traffic', in 2016 IEEE Security and Privacy Workshops (SPW), IEEE, 2016, pp. 245–251. Accessed: Jan. 04, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7527776/

[30] P. Yadav, Q. Li, R. Mortier, and A. Brown, 'Network service dependencies in commodity internet-of-things devices', in Proceedings of the International Conference on Internet of Things Design and Implementation, Montreal Quebec Canada: ACM, Apr. 2019, pp. 202–212. doi: 10.1145/3302505.3310082.

[31] A. Hamza, H. H. Gharakheili, and V. Sivaraman, 'IoT Network Security: Requirements, Threats, and Countermeasures'. arXiv, Aug. 21, 2020. doi: 10.48550/arXiv.2008.09339.

[32] Kamaldeep, M. Dutta, and J. Granjal, 'Towards a Secure Internet of Things: A Comprehensive Study of Second Line Defense Mechanisms', IEEE Access, vol. 8, pp. 127272–127312, 2020, doi: 10.1109/ACCESS.2020.3005643.

[33] P. I. Radoglou Grammatikis, P. G. Sarigiannidis, and I. D. Moscholios, 'Securing the Internet of Things: Challenges, threats and solutions', Internet of Things, vol. 5, pp. 41–70, Mar. 2019, doi: 10.1016/j.iot.2018.11.003.

[34] H. Gupta and S. Sharma, 'Security Challenges in Adopting Internet of Things for Smart Network', in 2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT), Jun. 2021, pp. 761–765. doi: 10.1109/CSNT51715.2021.9509698.

[35] V. Sivaraman, D. Chan, D. Earl, and R. Boreli, 'Smart-Phones Attacking Smart-Homes', in Proceedings of the 9th ACM Conference on Security

& Privacy in Wireless and Mobile Networks, Darmstadt Germany: ACM, Jul. 2016, pp. 195–200. doi: 10.1145/2939918.2939925.

[36] MITRE, 'NVD - CVE-2018-7580', NVD - CVE-2018-7580. Accessed: Aug. 27, 2023. [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2018-7580

[37] A. Oliveira, D. Fiser, and M. Logan, 'Endpoint Denial of Service, Technique T1499 - Enterprise | MITRE ATT&CK®', Endpoint Denial of Service. Accessed: Jun. 22, 2023. [Online]. Available: https://attack.mitre.org/techniques/T1499/

[38] Kali Linux, 'hping3 | Kali Linux Tools', Kali Linux. Accessed: Sep. 14, 2023. [Online]. Available: https://www.kali.org/tools/hping3/

[39] J. Touch et al., 'Service Name and Transport Protocol Port Number Registry', Service Name and Transport Protocol Port Number Registry. Accessed: Aug. 30, 2023. [Online]. Available: https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml?search=1&skey=4&page=54

[40] IANA, 'Service Name and Transport Protocol Port Number Registry', Service Name and Transport Protocol Port Number Registry. Accessed: Aug. 30, 2023. [Online]. Available: https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml?search=11095#Nest_Labs_Inc

[41] S. Notra, M. Siddiqi, H. H. Gharakheili, V. Sivaraman, and R. Boreli, 'An experimental study of security and privacy risks with emerging household appliances', in 2014 IEEE conference on communications and network security, IEEE, 2014, pp. 79–84. Accessed: Jan. 14, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6997469/

[42] S. Shirali-Shahreza and Y. Ganjali, 'Protecting home user devices with an SDN-based firewall', IEEE Transactions on Consumer Electronics, vol. 64, no. 1, pp. 92–100, 2018.

[43] IPFire, 'Raspberry Pi 4 Model B - The IPFire Wiki', Raspberry Pi 4 Model B. Accessed: Sep. 19, 2023. [Online]. Available: https://wiki.ipfire.org/hardware/arm/rpi/four

[44] IPFire, 'Firewall Default Policy - The IPFire Wiki', IPFire Wiki. Accessed: Sep. 19, 2023. [Online]. Available: https://wiki.ipfire.org/configuration/firewall/default-policy

[45] IPFire, 'IP Address Blocklists - The IPFire Wiki', IPFire Wiki. Accessed: Sep. 19, 2023. [Online]. Available: https://wiki.ipfire.org/configuration/firewall/ipblocklist

[46] IPFire, 'Location Block - The IPFire Wiki', IPFire Wiki. Accessed: Sep. 20, 2023. [Online]. Available: https://wiki.ipfire.org/configuration/firewall/geoip-block

[47] Suricata, 'Suricata User Guide', GitHub. Accessed: Sep. 20, 2023. [Online]. Available: https://github.com/OISF/suricata/blob/master/doc/userguide/what-is-suricata.rst

[48] IPFire, 'Intrusion Prevention System (IPS) - The IPFire Wiki', IPFire Wiki. Accessed: Sep. 16, 2023. [Online]. Available: https://wiki.ipfire.org/configuration/firewall/ips

[49] IPFire, 'Status - The IPFire Wiki', IPFire Wiki. Accessed: Sep. 20, 2023. [Online]. Available: https://wiki.ipfire.org/configuration/status