

Hossam Mohamed

# Internet Home Security

Helsinki Metropolia University of Applied Sciences

Master of Engineering

Information Technology

Master's Thesis

26 November 2019

Author(s)	Hossam Mohamed
Title	Internet Home Security
Number of Pages	53 pages
Date	26 November 2019
Degree	Master of Engineering
Instructor(s)	Ville Jääskeläinen, Principal Lecturer
<p>The concept of 'Internet of Things' is gaining fast momentum as multiple devices are being connected across home networks. However, ensuring the security of the home network remains a major concern as devices connected on the network are vulnerable to cyberattacks. Thus, the scope and aim of this research lie in outlining the potential security risks present in smart homes; and therein providing an affordable solution that can mitigate and reduce the threats posed by Internet intruders.</p> <p>In order to support the security of the home network against vulnerabilities and threats, research on Raspberry Pi Emulator (Desktop Edition) was conducted. This assisted in understanding the features that can be applied to protect the home network infrastructure against intruders and attacks. The proposed methods and materials were evaluated within predefined user and system requirements, which included several security threats and vulnerabilities.</p> <p>As a result, the following network security functions were successfully tested: Anti Virus – Files Scanning; Firewall; Malware Scanner; Intrusion Prevention System (IPS); Intrusion Detection System (IDS); and Ads Blocker. Although the study is bound by certain limitations, the research requirement of securing the network by behaving as an identity and alerting system, was nonetheless successful.</p>	
<b>Keywords</b>	Home Network Security, Intrusion Detection System, Intrusion Prevention System, Network Monitoring

## Contents

List of Figures

List of Abbreviations

1	Introduction	1
1.1	Background	2
1.2	Research Objective	5
1.3	Research Scope and Design	5
2	Literature Review	7
2.1	Security Overview	7
2.1.1	Security Fundamentals	7
2.1.2	Common Home Security Issues	8
2.1.3	Security Attacks	10
2.2	Home Network Security	14
2.2.1	Network Firewall System	14
2.2.2	Network Security Monitoring	15
2.2.3	Intrusion Detection System	16
2.2.4	Intrusion Prevention System	17
2.2.5	Anti-Virus & File Scanning System	20
2.2.6	Malware Hash Registry Scanner	20
2.2.7	Web Advertisement Filters	20
2.3	Network Home Security Requirements	20
2.3.1	Single Board Computer	21
2.3.2	Network Packet Analysis - Application-Based	21
2.3.3	Network Switch Port Mirroring	22
3	Implementation & Installation	24
3.1	Solution Architecture	24
3.2	Installation & Configuration	25
3.3	VirtualBox Manager	25
3.4	Raspberry Pi OS	26
3.5	Antivirus & Firewall	30
3.6	RootKit & Malware Scanner	32
3.7	Intrusion Detection & Prevention System	34
3.8	Network-Wide Ad Blocking	36

4	Post-Implementation Analysis	43
4.1	Intrusion Prevention & Detection Tests	43
4.2	Anti-Virus & Malware	46
4.3	DNS Sinkhole & Advertisements Blocker	48
5	Conclusions & Future Considerations	52
	References	

## List of Figures

Figure 1. IoT threat and attacks schema that surrounds us.....	4
Figure 2. Information Security Triad.....	8
Figure 3. Man in the Middle attack.....	11
Figure 4. Session Hijacking Phase 1.....	12
Figure 5. Session Hijacking Phase 2.....	12
Figure 6. Illustration of DNS Spoofing attack.....	13
Figure 7. Illustration of brute force attack.....	13
Figure 8: Firewall acts as barrier.....	15
Figure 9. Blocking, filtering and denying mechanisms.....	16
Figure 10. IDS and IPS location in the network.....	19
Figure 11. Single Board Computer – Raspberry PI.....	21
Figure 12. Uses of Packet Analyzer.....	22
Figure 13. Port Mirroring.....	23
Figure 14. Home Network Security Architect.....	24
Figure 15. Virtual Machine environment.....	25
Figure 16. Creating and setting up the new OS on VirtualBox.....	26
Figure 17. Setup the OS and operation requirement & modify the image setting.....	27
Figure 18. Adding the Raspberry Pi image to the storage.....	27
Figure 19. Graphical install process & loading additional components.....	28
Figure 20. Guided Disk Partitioning.....	28
Figure 21. Guided Disk Partitioning, installing services and copying to disk.....	29
Figure 22. Finish the installation and restart.....	29
Figure 23. Updating and upgrading the Raspberry Pi Package.....	30
Figure 24. Installing ClamAV Antivirus and updating the signature DB.....	31
Figure 25. ClamAV scanning the Operating System.....	31
Figure 26. Installing UFW Firewall.....	32
Figure 27. Installing rkhunter Rootkit package.....	32
Figure 28. Performing rkhunter scanning.....	33
Figure 29. Rkhunter system checks summary.....	33
Figure 30. Installing BriarIDS GUI.....	35
Figure 31. Installing BriarIDS GUI.....	36
Figure 32. Cloning into Pi-Hole.....	37
Figure 33. Raspberry Pi transformed into network-wide adblocker.....	38
Figure 34. Setting up Pi-Hole.....	38

Figure 35. Pi-Hole Third Party Services and TCP/IP protocols. ....	39
Figure 36. Configure management IP address and install web admin interface. ....	40
Figure 37. Pi-Hole installation process.....	40
Figure 38. Pi-Hole setup complete.....	41
Figure 39. Pi-Hole web Dashboard.....	41
Figure 40. Pi-Hole Database and signature updates.....	42
Figure 41. Executing the TCP replay for malicious traffic.....	43
Figure 42. The Squert Alerting Dashboard.....	44
Figure 43. Signature information and Source/destination address.....	45
Figure 44. Anti-Virus scanning patterns.....	46
Figure 45. Anti-Virus test file in multiple extensions.....	47
Figure 46. Access Denied for the EICAR-Test File.....	48
Figure 47. Pi-Hole Dashboard.....	49
Figure 48. Pi-Hole Blacklist gravity.....	49
Figure 49. Pi-Hole Dashboard after updating Domains on Blacklist.....	50
Figure 50. Pi-Hole testing websites for ad-block feature.....	50
Figure 51. Pi-Hole Recent Queries.....	51

## List of Abbreviations

ACL	Access Control List
ARP	Address Resolution Protocol
ASCII	American Standard Code for Information Interchange
AV	Antivirus
CPU	Central Processing Unit
DNS	Domain Name System
DoS	Denial of Service
DHCP	Dynamic Host Configuration Protocol
EICAR	European Institute for Computer Antivirus Research
EPEL	Extra Packages for Enterprise Linux
GPL	General Public Licence
GUI	Graphical User Interface
HIDS	Host Intrusion Detection System
IoT	Internet of Things
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IPsec	IP Security
IDE	Integrated Development Environment
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
MAC	Media Access Control
NIDS	Network Intrusion Detection Systems
NSM	Network Security Monitoring
OS	Operating System
Rkhunter	Rootkit Hunter
SSH	Secure Shell
SSL	Secured Sockets Layer
SBC	Single Board Computer
TCP	Transmission Control Protocol
UFW	Uncomplicated Firewall
UTM	Unified Threat Management
VLAN	Virtual Local Area Network
VM	Virtual Machine
VoIP	Voice over Internet Protocol

## 1 Introduction

The Internet has become the backbone of life for many people, and not just for businesses and technology experts. As there is a continuous rise in usage of Internet messaging services and the widespread usage of Voice over Internet Protocol (VOIP), the significance of the Internet for day to day lives cannot be ignored. For example, in some instances, individuals require the Internet for business purposes, while others use it as a communication method to connect with relatives and friends around the world; it is further used as one of the essential channels of entertainment. However, heavy reliance on the Internet also exposes users to vulnerability in many forms. One such example is botnet, which is a group of networks connected to devices and appliances via the Internet, contaminated with spyware and vicious malware, which in turn violate users' permissions by compromising their control. In the past, botnet attacks have revolved mainly around interrupting the services for blackmailing and ransom purposes. Therefore, it is imperative that the safety and security of home networks be improved in order to protect users from such threats.

Additionally, the concept of the 'Internet of Things' (IoT) is also becoming popular in this era of smart technology. Typically, IoT refers to any device, irrespective of size, use, or form factor, that consists of a Central Processing Unit (CPU) and memory, run software, and has a network interface which enables communication with other devices, usually as a client, and sometimes as a server (Stanislav and Beardsley, 2015). The purpose of IoT is to serve and automate the services for home users, improving the first version of the Internet and making it more powerful to manage and operate most of the elements of day to day life. In addition, these 'things' usually do not resemble traditional computers as they lack the typical keyboard and mouse interfaces, and their user interface is also not centered around a monitor or similar text-filled screen. Moreover, these 'things' are marketed and treated as single-purpose devices, instead of the general-purpose computers they are.

As there is an ever-growing network of physical objects requiring Internet connectivity, the use of IoT is significantly expanding among residential users across the globe. For instance, in the modern world, almost every household possesses more than three Internet-connected gadgets and devices that are connected to each other and to the Inter-



net. This is possible because the infrastructure that handles the data transmission protocols via the network has become smooth, such that it is now easy to connect to various vendors as well as different types of devices, gadgets, and appliances. Nonetheless, ensuring the home network's security is one of the main concerns of householders and the majority of the home network users. However, the security of the transferred data from and to an external network and the Internet is not given due consideration. Therefore, lacking visibility and transparency of communication is considered as one of the critical aspects of an in-home network wherein the traffic flow is invisible unless it is specifically searched for.

In addition to the above, security challenges related to user's data and privacy are also increasing as they are threatened to be hacked and exposed on the Internet. For instance, such attacks tend to target data and information in exchange for ransom or other such threats. Thus, to facilitate transparency, the security services and products installed in home networks need to address various challenges, such as product licensing, limitations of the features, not meeting the minimum requirements and missing support to be installed on a particular operating system (OS).

Additionally, home networks have become crucial as they form the center of operation that connects many smart devices, electronics and cooking appliances - both indoor and outdoor. For instance, home security services such as CCTV cameras, motion sensors and smoke detectors connected in these networks are also vulnerable towards outside attacks. Such vulnerability raises several questions such as: how efficient are the current type of security measures installed in homes and can they protect users from Internet attacks? Moreover, are these measures able to secure users' data and privacy?

## 1.1 Background

Regardless of the existence of various options in the market that provide protection and detection for home network security, home networks are not sufficiently secured to the level where they can smartly interact with all aspects and dimensions of a cyberattack, especially when the attack is distributed on several devices which further complicate it. Furthermore, utilizing a home network security without proper security settings and protection can cause numerous exposures that eventually would lead to vulnerability. As such, managing the home network's security infrastructure has been a perpetual challenge for the majority of home users. Thus, even though the variation of smart devices used on a daily basis is beneficial to most users, it, however, raises security

concerns when it comes to threats and vulnerabilities. Moreover, users' act of ignoring such security measurements is on the rise and, therefore, the need to protect home network infrastructure is imminent. Thus, this forms the basis of the problem statement addressed by this research, further detailed below.

A major function of home networks is that it does not only allow access to devices to the Internet and external networks but also shares information. Both of these provide advantages on one hand and cause security challenges on the other. Therefore, this section points out some of the home network security challenges as below:

- i. Lacking home network traffic visibility and control
- ii. Absence of fully automated protection and defensive system
- iii. Lack of awareness of home network attacks and threats
- iv. Free and affordable home network security solutions not as widely used as commercial ones

Apart from this, the common threats in-home network are malware, virus, and spyware that can be reformed into several shapes and forms such as email attachments and tiny executable software, which are activated automatically. For this, the targeted core of a home network is usually the home network gateway. Additionally, the cyberattack may intend to manipulate the integrity of the data or its availability on the system, and even damage it by hacking into the weak core of the system. Therefore, security specialists are looking for solutions to constrain the episodes causing perceptible harm, by recognizing it from the more scheduled information breaches and attack activities.

Although many users and industrial companies find it to be convenient to connect all their devices and appliances, which share the same type of communication, it opens up a possibility of attacks from unknown entities. Thus, one should take precautionary measures to protect their assets. Moreover, the power of the Internet and computers, and the inability to secure them in the best of circumstances makes the problem far worse today (FireEye, 2018). For instance, as an increasing number of IoT devices are being deployed in homes now, which are encouraged to be connected to the Internet, cyber offenders have more desire to leverage these devices in their attack (see Figure 1). This is especially true now that there are more IoT devices on the market, such as wearables, pacemakers, thermometers, and smart plugs. This heightened interest is due to the vulnerabilities in many IoT devices, not to mention their ability to connect, which

can form an IoT botnet (Sarang, 2018). For example, most of the IoT gadgets, such as the home router device can be manipulated to serve the purpose of botnet assaults. This device, in particular, is the heart of the Internet in homes as it is the gathering point for all appliances and smart devices like smartphones, smart TVs, radio, lights, home locks, baby monitors, etc. Apart from the home router, the other connected devices are vulnerable to digital cyber hackers as well, an example of which is discussed below using the case study of baby monitors.

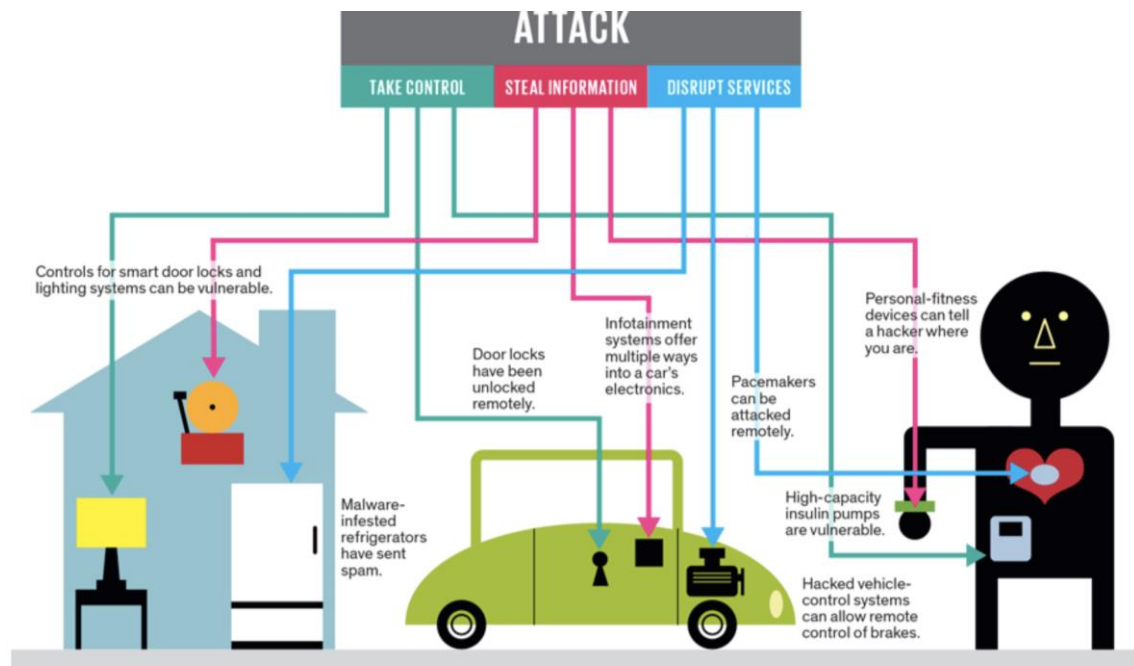


Figure 1. IoT threat and attacks schema that surrounds us.

Rapid 7 (a prestigious vendor in the security solution world) conducted research on different Internet Protocol (IP) video baby monitors. Their study focused on the protection and security of such monitoring devices installed in smart homes from a different perspective. That is, small baby monitors are usually set up close to babies and infants to provide parents with peace of mind while they are away and/or when nannies take care of them. These devices are known for their high privacy settings and the durability of transmitting data (through voice and video streaming). Even though these devices are small in size, family members are given access to watch their home while traveling, which can ensure the safety of their home and belongings. Although baby monitoring devices are an ideal example of security reconnaissance as explained above, they are not marked as secure or protected devices. The study by Rapid 7 exposed several known as well as new vulnerabilities in the design of such devices making up the IoT.

Moreover, it is argued that these findings not only apply to video baby monitors but also provide a good sense of what consumer IoT security looks like in general (Ashford, 2015). From this, it is evident that the mechanism of finding these gadgets is universal and easily transferable to other interests, such as products that use the same techniques as automation systems and climate control systems. Hence, the problem of the evergrowing IoT network is that it can open up a gateway for cyber attacks due to security breaches, thus laying the foundation for the scope of this paper.

## 1.2 Research Objective

The scope of this research lies in outlining the potential security risks present in smart homes and therein, providing an affordable solution that can mitigate and reduce the threats posed by Internet intruders. As such, this paper aims to discuss and explore the dimensions of security threats when communicating with other devices and appliances via the Internet, and further attempts to spread security awareness; these aspects are outlined in chapter 2. Furthermore, based on this literature review, the mitigation of the Internet and IoT threats for home users nowadays are subsequently discussed. For instance, different security vendors have issued white papers to discuss the threats of the Internet and IoT that are currently active in homes, including countermeasures and impact. Nonetheless, due to the inadequacy of the current measures that give rise to the scope of this research, the following research questions were formulated, which helps define the solution to the problem:

- Q1. What defense mechanisms are in place to protect users from such threats?
- Q2. Is there visibility on Internet traffic as well as transparency on inbound/outbound data flow in-home network?

## 1.3 Research Scope and Design

The main focus of the paper is to propose an affordable home security solution covering most of the security features for home users. A proof of concept is discussed hereunder to gain an actual overview of this proposed solution, which will help answer the research questions stated above. Moreover, the literature reviewed advocate the importance of securing the home network and discuss network vulnerabilities as well as threats that could accidentally happen due to a lack of security in-home network infrastructure. Moreover, to support home network security against vulnerabilities and threats, research

on Raspberry Pi Emulator (Desktop Edition) is conducted, which further assists in understanding the requirements of features that can be applied to protect the home network infrastructure against intruders and attacks.

The proposed methods and materials are evaluated within predefined user and system requirements, which includes several security threats and vulnerabilities. In addition, a virtual machine with Raspberry Pi OS was downloaded and installed to simulate the network security measures solution. The scope of the study includes the following network security functions:

- i. Anti Virus – Files Scanning
- ii. Firewall
- iii. Malware Scanner
- iv. Intrusion Prevention System (IPS)
- v. Intrusion Detection System (IDS)
- vi. Ads Blocker

This paper is divided into five sections, the first and foremost of which is the introductory chapter that provides the problem statement and outlines the aims and methodology of the research, in addition to discussing the study background. The second chapter, 'literature review,' explains the theoretical concepts underpinning the study, which is further written in three parts by firstly giving a general overview of security, and therein detailing home network security features and home network security requirements. The third chapter focuses on the 'implementation' of the proposed solution, which is followed by the 'analysis' chapter that validates the performance of the proposed solutions. Finally, 'conclusions' are drawn in the last chapter that also discusses limitations and outlines a possible future study direction.

## 2 Literature Review

This chapter outlines the theoretical background underpinning the study by first discussing the issues surrounding home security in section 2.1. Secondly, a review of the functions performed by home network security is provided in section 2.2. Finally, the basic components required for implementing home network security solutions are discussed in section 2.3.

### 2.1 Security Overview

The fundamentals of security are first explained below, followed by a discussion of the elements that affect it, such as the types of misuse, human acts and external factors threatening security. In addition, an evaluation of the impacts and risks is conducted pertaining to the vulnerabilities and threats faced by home networks. Furthermore, the level of exposure is also taken into consideration, in addition to the behavior of the intruder and the motivation behind the attacks.

#### 2.1.1 Security Fundamentals

Stolfo (2011) argues that in order to understand security, the impairments to it must first be understood. For instance, similar to fault tolerance, the correct behavior of systems can only be assured when the reasons for failure can be identified, such as why and how the systems would fail. The author further argues that there are two leading causes of system failures, as listed below:

- i. Attacks or faults maliciously made by humans
- ii. Vulnerabilities or unintentional faults made by designers.

Of these, the latter termed as 'vulnerability' is the root cause of failure in security measures made by humans unintentionally; hence, addressing such failures can assist in establishing more secure systems. Generally, the Information Security Triad is used as a guide for organizations that highlights the core data security objectives: confidentiality, availability and integrity (see Figure 2).



Figure 2. Information Security Triad.

The Information Security Triad given in Figure 2 is a benchmark model that evaluates the information security of an organization. Moreover, to ensure adequate security posture for organizations from different sectors, the three elements of the Security Triad, which stand for 'confidentiality,' 'integrity' and 'availability,' should be maintained well to suit their operational needs. For example, banks consider 'integrity' to be the most crucial element, whereas a data processing company would consider 'availability' to be the most important one.

### 2.1.2 Common Home Security Issues

The most common home security issues that are faced by home users concerning devices such as laptops, desktops, smart mobile devices, as well as IoT devices, are discussed hereunder.

The typical workings of WiFi routers and connected devices are as follows: by using the default network router gateway in local networks, traffic flow is directed to the desired destination. However, in the absence of strong methods for authenticating devices to wireless networks (such as lack of certificate-based authentication), an attacker can exploit the situation and gain access to the wireless network. In order to avoid such threats, the default credentials should be changed to a complex format. For instance, home devices and appliances are connected to the single default gateway, and routers are manufactured with a pre-issued, self-assigned certificate. However, some routers have untrusted certificates that cause vulnerabilities. For instance, man-in-the-middle attacks (explained in section 2.1.3.3) rely heavily on these untrusted certificates as they are easy to replicate. Thus, to avoid this, one can use a trusted Secured Sockets Layer (SSL) certificate obtained from Certificate Authority.

A second common issue found among home networks is the use of default credentials for the router web management interface. Because of this, an attack of privilege escalation can easily take place, which is considered a network-type intrusion. This intrusion takes advantage of the laid back attitude of users to grant the hackers access that compromises the network and application credentials.

A third collective security issue that all home users usually suffer from is a lack of transparency in Internet traffic. This is because users cannot be certain that all the services running on the home network are indeed services initiated by them in the first place, as it could also be applications mistakenly granted access to the Internet. These unauthorized applications running in the network such as Virus, Trojan, Malware, and Spyware have many forms today – all of which require an Internet connection to connect the intruder to the victim in order to gain access to personal data and information; following this, user threats such as ransom attacks may also be initiated.

Furthermore, most of the smart home devices of today and IoT can attract different types of risks that the intruder targets, for example, the baby monitoring system discussed previously. For instance, the storage service for most baby monitoring systems is located on the Cloud, whereas a few also store locally on memory units in the device itself. For the attackers to be able to execute the attack, they will be required to sniff the local traffic to obtain the media access control (MAC) address of the baby monitor and acquire Camera IDs. Moreover, the attackers usually tend to have a logical technique that is simple and easy to use in the first stage. For example, by knowing the vendor's name or model of the baby monitoring device, they are likely to try the default credentials of the manufacturer (username and password) to gain access. As such, information gathering and analysis are considered as one of the penetration stages that most attackers use. This is done by scanning tools that are used commonly to understand and discover types and responses of targets of intrusion.

Nonetheless, the motive behind attacks may vary. In most cases, the intruders seek to prove intellectual superiority over victims, who are not as skilled in protecting their own networks and devices from such attacks. However, these victims are left with nothing but security measures to defend themselves. These measurements, from the perspective of an average home user, tend to be expensive as security vendors often take a commercial approach for their security services and products. Moreover, all security appliances and software are limited to the licenses and validity of the registration.



Nonetheless, the cost of these services is minimal compared to the value of losses that can result from such attacks, e.g., theft of property insurance policy, etc.

### 2.1.3 Security Attacks

The shape of attacks tends to differ from one device to another. For example, the software source code can be injected into a specific operating system (Android/Windows /Linux /Unix/MAC OS) from the malicious application of the unverified application publisher. However, security concerns put more focus on preventing unauthorized access and invading privacy as well as altering the integrity of data. As such, this section covers the most common invasive attacks that target information, network and personal data; hence, different types of techniques and methods used to carry out successful attacks are reviewed below.

#### **Denial of Service (DoS)**

DoS attack aims to disrupt the service temporarily by making the service and network resources unavailable to its intended users. The mechanism of this attack is summarised in exploiting the vulnerability of operating systems or services. As such, it can cause the immediate shutdown of the operating system, which leads it to crash; soon enough, the system becomes inaccessible. Usually, the attacker triggers the attack from many locations at once. In addition to this, such kind of attacks comprise of specific features, i.e.:

- The attack cannot be determined easily from a location perspective, as the IP address changes due to the various distribution of the attacking system.
- The attacker can manipulate various machines under the same network aiming to cause damage or crash the entire system.
- Attacker identity is usually anonymous and fabricated under another third party, which makes it difficult to reveal.

#### **TCP Flood Attack**

This attack is often used to buffer space during the utilization of the main network protocol, which is the Transmission Control Protocol (TCP) session in handshake operation. Herein, the system is usually flooded by tiny connection requests and – by default – it is not capable of handling such high amounts of request; in turn, this causes the victim's system to time out. Meanwhile, the attacker awaits a response from the

victim's device, which results in inaccessibility or system crash, and thus prevents reachability for users and other systems.

### Man in the Middle Attack

Generally, attackers attempt to alter the communication between two parties such as client and server in secrecy, by intercepting the victim's communication. As such, Address Resolution Protocol (ARP) spoofing is one of the standard tools used to redirect traffic in the victim's network to the attacker's network. Hence, the main goal of the man-in-the-middle attack is to steal personal data and control the victim's system to redirect it to meet the attacker's targets. Moreover, the attacker usually uses a specific strategy to achieve such an attack, as shown in Figure 3.

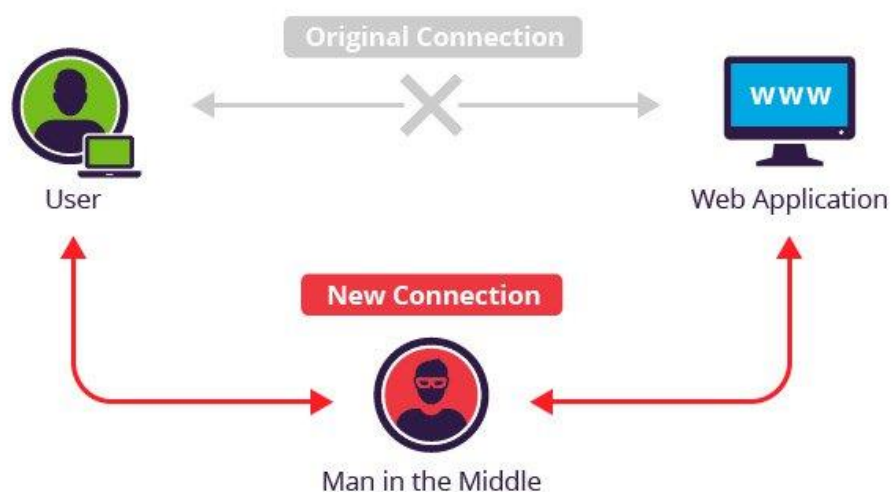


Figure 3. Man in the Middle attack.

In addition to the above, the attackers can reach their target destination by intercepting their victim's traffic using a passive attack like fake identical search engines, or use of popular websites, poisoned WiFi hotspots, lottery emails with the detailed filling application, etc. Once the victim accesses any of these vicious traps, the attackers can gain full accessibility and visibility of the victim's data. Contrastingly, the attackers can also seek to dominate their victim's traffic using an active approach by launching the following attack methods:

- i. IP Spoofing: The attacker modifies the packet header in the IP address in the form of tiny software or application. Once the victim accesses this application, the attacker will have full accessibility and control over the victim's machine; this is termed as 'Session Hijacking,' as shown below in Figures 4 and 5.

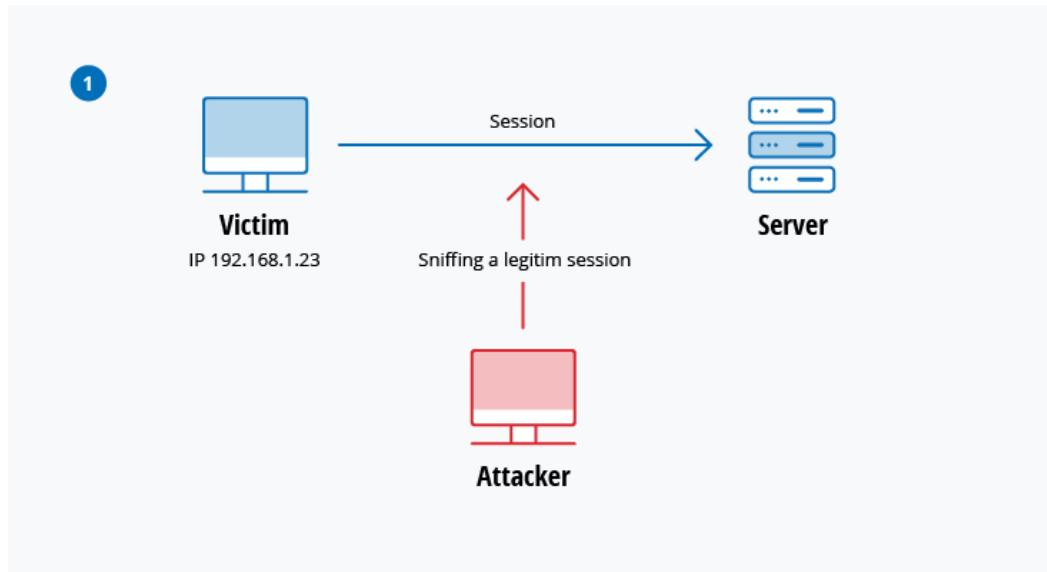


Figure 4. Session Hijacking Phase 1.

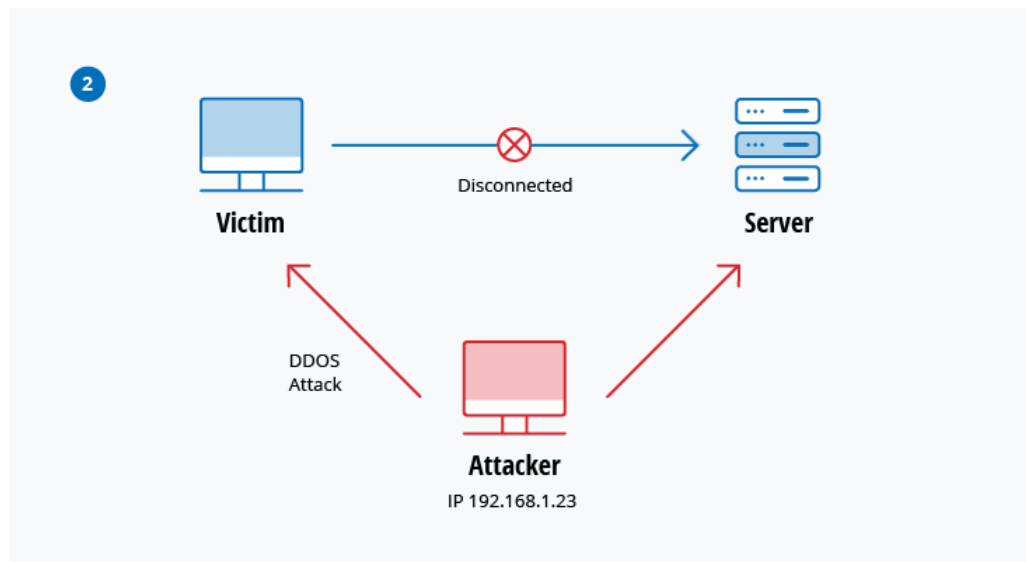


Figure 5. Session Hijacking Phase 2.

- ii. **DNS Spoofing:** Domain Name System (DNS) spoofing occurs when the attackers redirect the victim's traffic to provide them with multiple options to harm the victim, such as deploying Trojan, keyloggers, and malware. When the victim tries to access the website where the DNS is altered, they get redirected to the attacker who gains access to all user credentials and data, as illustrated below in Figure 6.

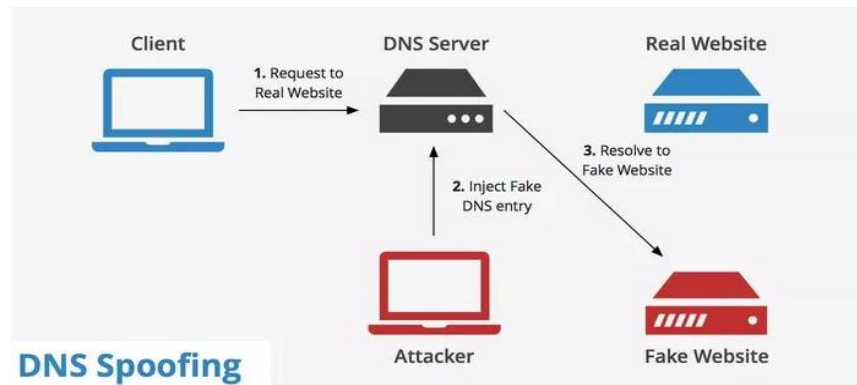


Figure 6. Illustration of DNS Spoofing attack.

- iii. Password Attack: Although more sophisticated and secure biometrics systems are used today, such as iris scan, fingerprint, face recognition, etc., finding passwords written on a piece of paper or note on the desk next to the laptop, can be an easy hunt for attackers. Apart from this, sniffing the password from network traffic is also an easy opportunity for attackers, especially if it was unencrypted. Moreover, using social engineering can be an effective method to impersonate victim identity as an automated password prediction process can consume time and CPU utilization to make it successful. In line with these, there are different methods and approaches to password attacks that help attackers acquire them from the victim's network. Among these approaches, attackers prefer specific methods such as brute force attack (as shown in Figure 7) and dictionary attack.

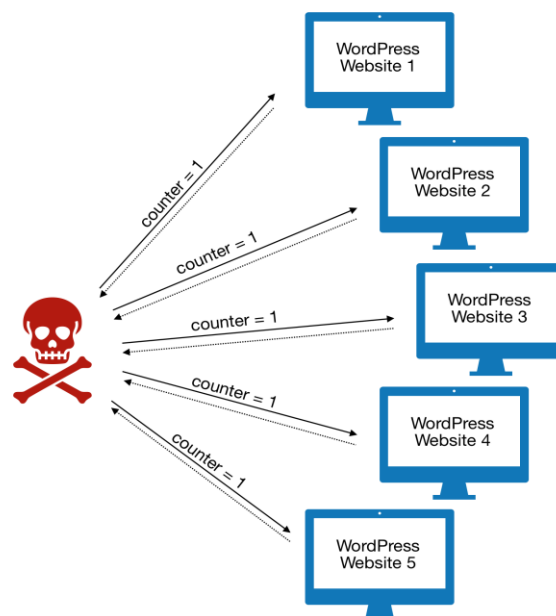


Figure 7. Illustration of brute force attack.

Within the brute force approach, password guessing and predicting valid passwords of active systems is attempted. For instance, a random approach is used by generating a massive amount of words, characters, numbers, and combinations, in consecutive attempts until the current password is detected. On the other hand, the dictionary attack method relies on a program that is loaded with a list of words (single or combined), numbers and symbols, with which the attacker attempts to gain access to the user's computer.

## 2.2 Home Network Security

In this section, the features of home network security and the functionality of each of the features are further discussed. As such, home network security is intended to be used for detecting and defending home security as well as provide network forensics to spyware and malware. As detailed above, conventional methods cannot offer full and comprehensive protection for home users today because the existing security mechanisms are usually incapable of defending users against all types of attacks. Moreover, the lack of tools and systems to inspect network traffic – outbound and inbound – prevents identifying the type of traffic, such as whether it is secure or dangerous to be allowed on the network. Additionally, missing the defensive shield of network security, such as Intrusion Detection System (IDS) and Intrusion Prevention System (IPS), tends to create a blind spot and a weakness gap between the assets requiring protection and the external network. Thus, to mitigate these issues, the following solutions are reviewed in this chapter, which can be used to address the problems faced by home users.

### 2.2.1 Network Firewall System

In the computing industry, the network firewall system is considered the first line of defense for the entire network and computing infrastructure that can inspect and monitor incoming as well as outgoing data traffic flow. As such, firewalls make a barrier between trusted and untrusted networks such as DMZ and the Internet (see Figure 8). Moreover, the function of a firewall system is to filter network traffic flow based on the configured policy (note: a policy includes a group of rules which indicate whether a specific type of traffic flow is accepted or denied).

## DMZ network architecture

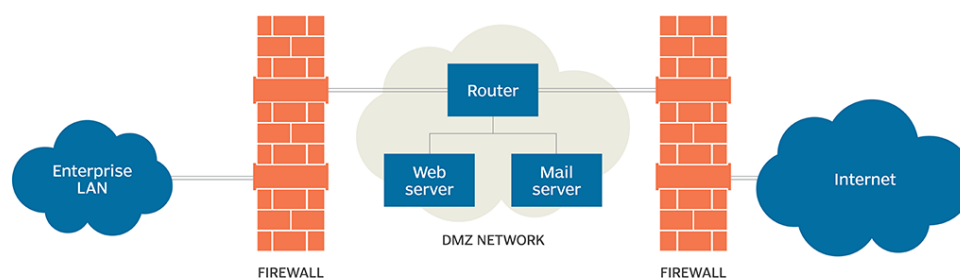


Figure 8: Firewall acts as barrier.

Furthermore, firewalls are categorized into two types – stateless and stateful – as explained below (Mojidra, 2016):

- Stateless Firewall is also recognized as an Access Control List (ACL). This type of firewall inspects traffic and evaluates packets' metadata. It further monitors and filters traffic packet payload and headers, including incoming and outgoing ones; therein, it performs actions based on matching criteria in the firewall policy.
- Stateful Firewall is a type of firewall that tracks the connection status to monitor traffic stream. It also recognizes communication paths and is capable of implementing IP Security (IPsec) functions like encryption. In other words, a stateful firewall monitors the status of the TCP connection, such as whether it is open, open sent, synchronised, synchronisation acknowledged or established.

### 2.2.2 Network Security Monitoring

One of the ways to keep a network secure is to know the network inside and out. For instance, network security monitoring can be carried out in many different ways, but all of the methods require awareness regarding what is going on in the network (Veríssimo and Rodrigues, 2009). Hence, the purpose of Network Security Monitoring (NSM) is to provide an overview of the network traffic flow and nature, and therein alert the owner of the network before an intruder causes any damage to their assets (Sanders, Smith and Bianco, 2014). Thus, NSM, for the most part, is responsible for detecting malicious traffic in an effort to take the best action against threats or attacks on the network. As such,

high CPU or network utilization is a good indication of high resource consumption, which usually indicates a vicious activity that requires investigation.

Bejtlich (2008) further argues that NSM, having its own set of characteristics, is a new paradigm for the detection domain; and in this regard, it differs from traditional intrusion detection methods. The authors also argue that prevention methods would ultimately fail against a motivated and dedicated attacker who will eventually be able to find a way in, regardless of how strong the defense methods are or what pro-active steps have been put in place. With regard to the operations of NSM, it does not usually block, defend, drop traffic or filter the IP/TCP packets. Nonetheless, its core functionality is to make the traffic more visible to users in case of failure of cybersecurity tools. For instance, tools such as IDS, IPS or Antivirus (AV) may fail to take proper action or behave as it should in the event of an attack, whereas NMS can serve as an alert tool to inform the user of an incoming intrusion.

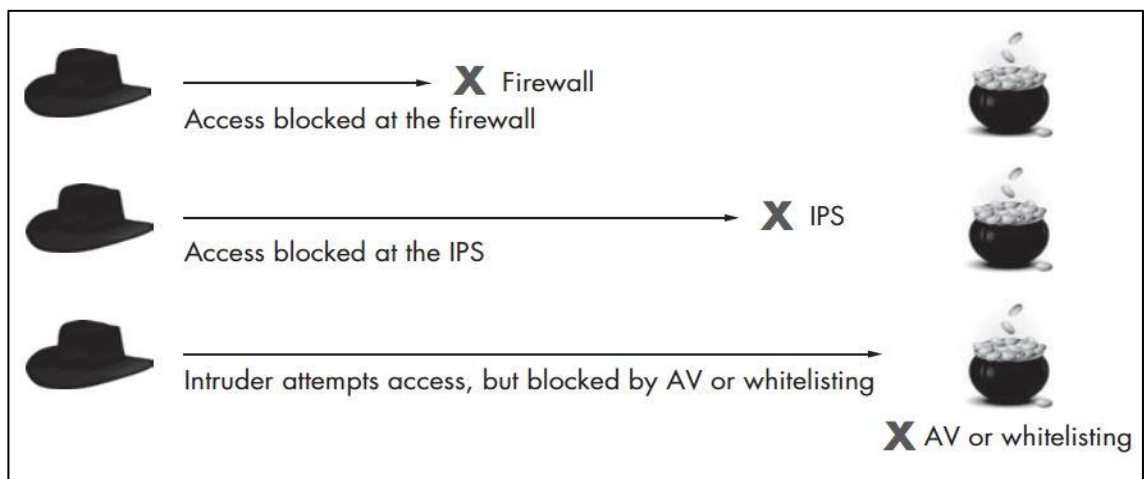


Figure 9. Blocking, filtering and denying mechanisms.

An example of the above is shown in Figure 9, where the usual defense systems recognize malicious activity and stop it; however, when such defense attempts fail, NSM can provide visibility of these failures.

### 2.2.3 Intrusion Detection System

Intrusion Detection Systems (IDS) are network appliances made to detect vulnerabilities as well as any suspicious events, threats or anonymity traffic. Once the IDS discovers these items, it sends alerts, indicating malicious activity, rule or policy violations. In its core, IDS can be classified as either passive or active. The passive type of IDS always

generates or triggers alerts once it detects suspicious activities without taking any actions. On the other hand, the active type of IDS carries out the same functionality of generating alerts and alarms, but also undertakes additional steps of taking action like blocking the suspicious IP address or dropping the malicious traffic that can cause harm to the system.

Apart from this, IDS can further be sorted into two types based on the detection mechanism applied, as explained below:

- i. Network Intrusion Detection Systems (NIDS) are deployed within the coverage to monitor traffic to and from all appliances and hosts in the network. Furthermore, NIDS audits and scans all passing traffic and classifies the type of traffic in each subnet; and then matches the nature of the traffic that is generated by each device and host within the network subnets. Once the attack has been sensed and abnormal activities have been detected, the alerts can trigger the system administrators.
- ii. Host Intrusion Detection System (HIDS) is defined as a local monitoring intrusion detection system that is located on the internal computer and has capabilities for analyzing network packets on its network's interface. Thus, HIDS provides an added benefit for the network to detect packets that are sourced from inside the computer, which NIDS may fail to recognize in the latter stage of detection. Moreover, being able to discover suspicious traffic that is generated from the host itself is an example of utilising resources well with a quick detection response.

#### 2.2.4 Intrusion Prevention System

An Intrusion Prevention System (IPS) is a type of network device that examines traffic flow and provides an in-depth investigation of the packet header and the entire data container to detect and prevent vulnerabilities from harming the network infrastructure (Forcepoint, 2019). As such, IPS actively monitors the network by searching for suspicious activities that cause malicious events. Therein, the generated alerts signal the network and system administrator to take preventive action regarding the finding of vulnerabilities.



In terms of functionality, IPS can integrate with a specific standard or policy, based on the user who can identify the structure of the type of traffic in use. Additionally, the core function of IPS is to scan the network traffic patterns and structure, forming a full network map; this occurs with the usual utilisation and nature of resources that require access to the Internet (TCP/IP ports) defined in the term to prevent particular types of attacks such as:

- Worms
- Denial of Attacks
- DDOS
- Zero-Day Exploits
- Virus
- Malware.

Furthermore, the IPS attains packet inspection in real-time, with a deep inspection of individual packets traveling through the network. Therein, on detection of suspicious or malicious packets, one of the below actions shall be performed by the IPS:

- Termination of exploited TCP sessions and blocking of offensive source IP addresses or user accounts from getting access to any application, targeting hosts or other networking resources.
- Reprogramming or reconfiguring the firewalls in order to avoid similar attacks from happening in the future.
- Removal or replacement of malicious contents which continue to remain on the network following any attack; this is performed through repackaging payloads, removing header data, and deleting corrupted or infectious attachments from files and mail servers (Veríssimo and Rodrigues, 2009).

The purpose of having IDS and IPS is to produce a comprehensive defence mechanism for full protection of the network infrastructure, as each these tools have a specific method of functionality. However, IDS is considered a passive system responsible for monitoring logs, notifications and alarms of the event, while IPS takes actions against what has been reported by the IDS. Thus, IPS is considered an active system that is responsible for taking action based on system architecture.

Moreover, as IDS is designed to differentiate between regular and suspicious traffic, false-positive on IDS is less likely generated on a network where the network administrator reviews traffic flows and patterns. Additionally, IDS has been developed to

quarantine the Zero Day attacks and other new malicious vulnerabilities that can compromise the network in less than milliseconds, which is considered a tremendously quick attack. Therefore, vendors and technology developers have considered merging this feature of IDS to be a part of the IPS.

In the current scenario, technology providers have released active IDS that can capture network traffic and take action based on its flow type. With time, it has become smarter and more capable of preventing suspicious attacks in the network. As such, both IDS and IPS have anomaly-detection features integrated. Nonetheless, IPS is usually set up and configured behind the firewall, providing an extra layer of analysis. Thus, it is located inline (between source and destination). Since its nature is active scanning and capturing of all traffic flow, it will take action based on the risk level, vulnerability and threat type.

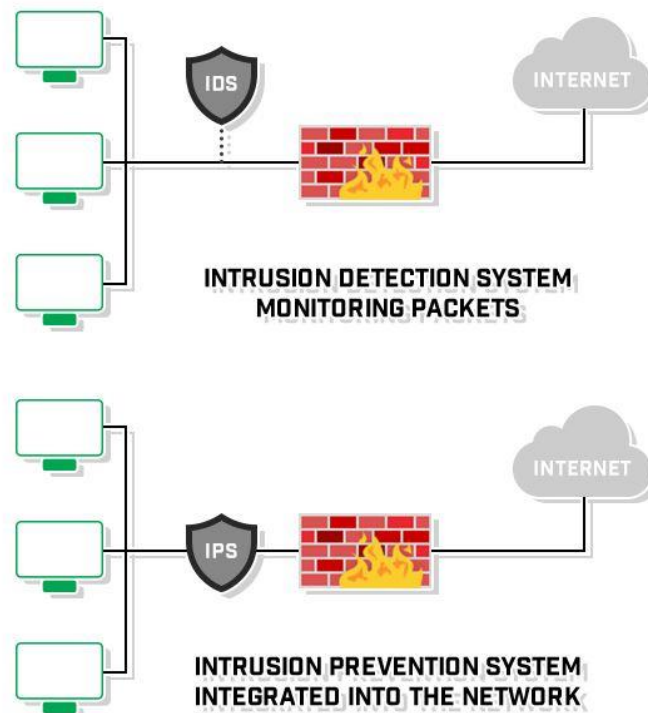


Figure 10. IDS and IPS location in the network.

As shown in Figure 10 above, when an IDS detects an intrusion, an alert is sent to the security/ webserver administrator to take action, whereas, an IPS located within the network acts one step further by automatically taking action to prevent the intrusion.

### 2.2.5 Anti-Virus & File Scanning System

An anti-virus scanner is a software that detects, prevents and removes viruses and Trojans.

In today's era, viruses have become more intelligent and developed, as viruses and malicious malware intend to compromise the information security triangle of confidentiality, integrity and availability. Nonetheless, modern intelligent antivirus software can secure home network users from several attacks and virus infections, such as:

- Spyware
- Trojan Horse
- Keyloggers
- Ransomware
- Phishing
- Browser Hijackers.

### 2.2.6 Malware Hash Registry Scanner

Malware hash registry scanner is a service that allows users to investigate and inspect file row data and metadata, including MD5 and SHA-1 hash of the suspicious signatures and identifications. As such, the malware hash registry scanner provides an added layer of security and detection.

### 2.2.7 Web Advertisement Filters

Web browsing is a regular feature that is more often used by home network users. However, threats such as browser hijackers are common, which attack Internet cookies and thus result in compromising the security of home user's data and privacy. For this purpose, applications that block Internet tracking can be used on private networks as these behave like a DNS sinkhole, or optionally as a Dynamic Host Configuration Protocol (DHCP) server. Moreover, these applications have been designed for usage on devices that are embedded with network capabilities (Verissimo and Rodrigues, 2009).

## 2.3 Network Home Security Requirements

This section describes three basic components that are needed for implementing the network security solutions that this paper advocates.

### 2.3.1 Single Board Computer

A Single Board Computer (SBC) is a small computing device that can be utilized for several purposes, such as in learning and development environments. As such, SBC consists of single board memory, input/output, a microprocessor and other features necessary for a functional computer. For example, the computer on the module is also a type of single boarded computer prepared for plugging into a backplane system expansion or a carrier board (Mojidra, 2016). Similarly, Raspberry Pi (see Figure 11) can be considered as a single board computer.

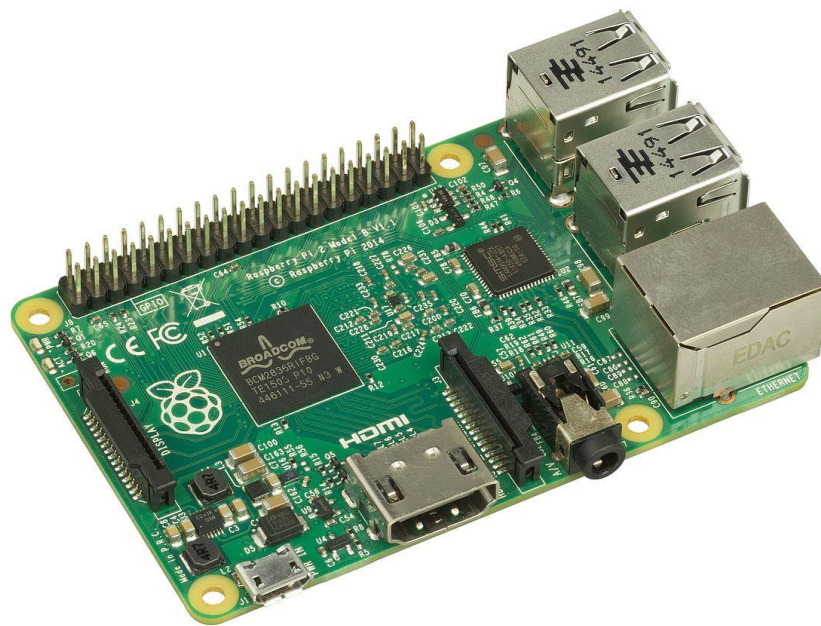


Figure 11. Single Board Computer – Raspberry Pi.

Raspberry Pi, as seen above in Figure 11, is a small compact size unit that plugs into a display monitor, and uses standard keyboard and mouse. This research has made use of a Raspberry Pi operating system to simulate the installation and functionalities of security measures that can be used as a home network security solution.

### 2.3.2 Network Packet Analysis - Application-Based

A packet sniffer is a computer software that intercepts network traffic. It can also perform packet capture, which is the process of grabbing the entire contents of packets or packet

headers. Capturing the packet header lower the storage required in the process. (Diamond Systems Corporation, 2017).

The Network Packet Analyzer is assigned to analyze raw data and network traffic flow to detect attacks and unusual traffic patterns. Thus, its sniffing feature can be used to provide enhanced network troubleshooting (Connolly, 2016).

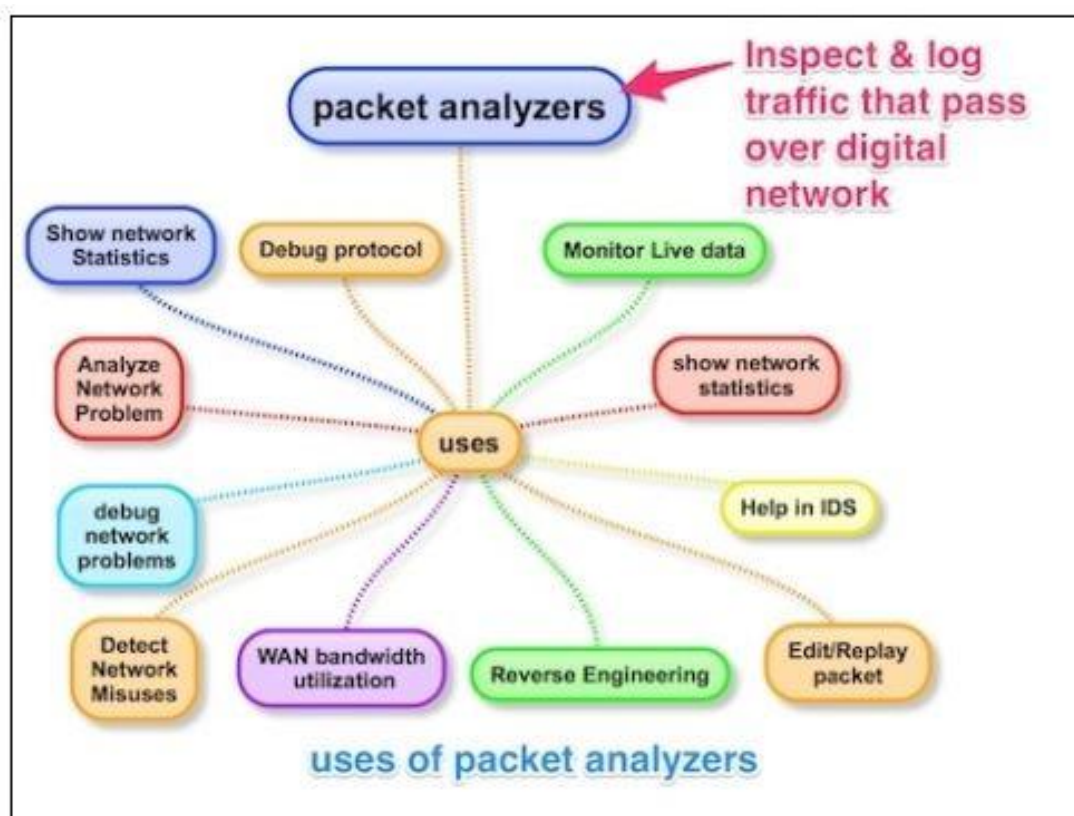


Figure 12. Uses of Packet Analyzer.

Figure 12 describes the many uses of a packet analyzer, such as analyzing network problems, detecting scans and attacks, and for sniffing, etc., by inspecting and logging the traffic that passes over the digital network.

### 2.3.3 Network Switch Port Mirroring

Port mirroring feature in network switch is used to provide an identical copy of network packets arriving at a certain physical interface or Virtual Local Area Network (VLAN) interface. To facilitate this, a software embedded feature built into the Network Switch creates a copy of packets passing via the switch to a specific port.

A network engineer frequently uses port mirroring for monitoring and analyzing purposes as well as for network administration and system administration. As such, port mirroring may be used to copy network data packets in the following cases (Juniper Networks, 2018):

- It can be used for all the packets exiting or entering any interface (in whichever combination). For instance, one could send packet copies entering or exiting certain interfaces to the same VLAN or local interface. If one configures port mirroring in order to make copies of packets exiting any interface, the traffic which originates over the switches or node devices (on QFabric systems) is not copied while it egresses; what gets copied on egress is only switched traffic.
- It can also be used for all the packets that enter a VLAN; however, one cannot choose port mirroring for copying packets that exit a VLAN.
- It can also be used for filtered firewall samples, i.e., packet samples that enter a VLAN or port. For this, one may need to configure firewall filters for choosing particular packets for mirroring.

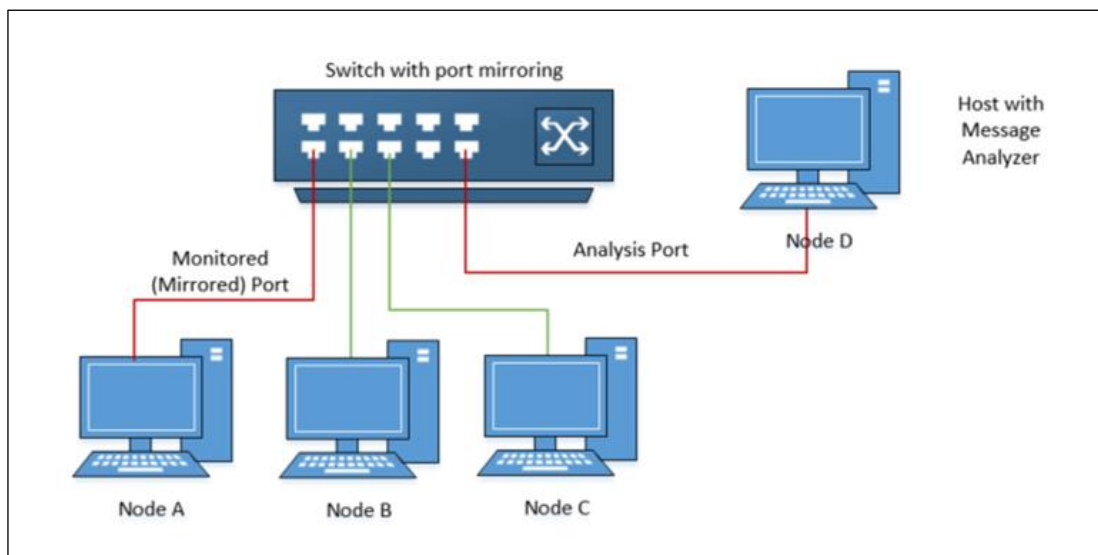


Figure 13. Port Mirroring.

Figure 13 above is an illustration of switch port mirroring, which uses a single egress port to aggregate multiple links.

### 3 Implementation & Installation

In this chapter, the implementation of the home network security solution is further detailed, which can also be replicated by other users and readers by following the steps described.

#### 3.1 Solution Architecture

Home network security is a collection of multiple systems that are designed to perform on a unified platform such as Raspbian and Debian. With regard to home network security monitoring and protection, the main aim is to provide an effective and affordable solution, as well as overcome the commercial licensing issue faced by home users.

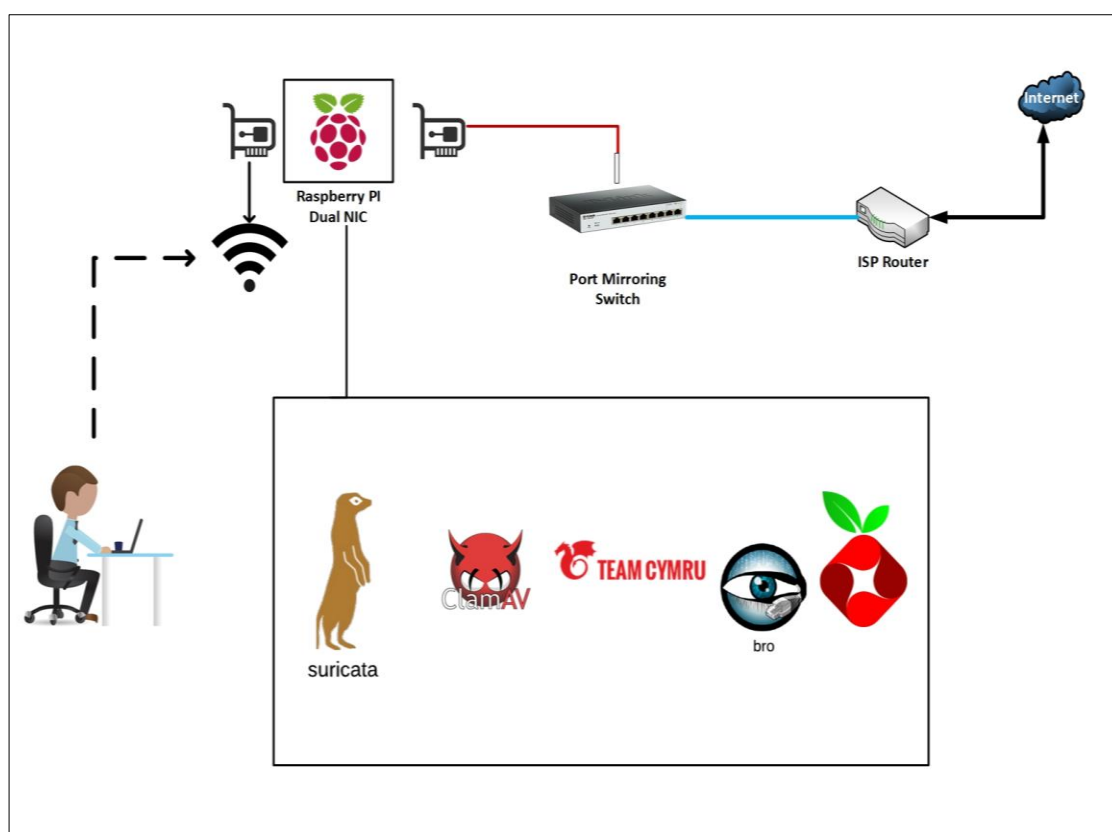


Figure 14. Home Network Security Architect.

The diagram in Figure 14 shows the architect of home network security containing port mirroring switch, Raspberry Pi SBC, Suricata etc., which are further explained in subsequent sections.



### 3.2 Installation & Configuration

To create a virtual lab to experiment on the desired environment, replacing the Raspberry Pi hardware with Virtual Machine (VM) is often recommended, as it provides virtual machine support to create an isolated testing environment. For instance, a VM with required Operating System image can simulate precisely the same functionality of real hardware, with added features such as isolating system from surrendering services and devices against potential threats and vulnerabilities.

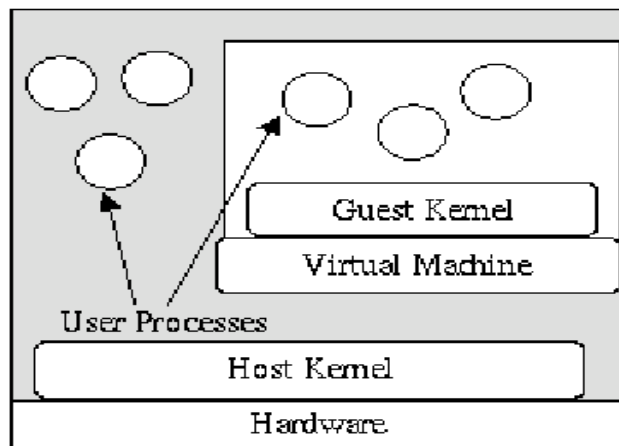


Figure 15. Virtual Machine environment.

Figure 15 above provides a visual representation of a VM environment, where the host kernel can be an OS such as Windows, while the guest OS can be the same or different OS such as Linux, and user processes such as Apps can run on either OSes.

### 3.3 VirtualBox Manager

VirtualBox users can load numerous guest OSes through a single hosting operational system (host OS). For instance, each of the guests can start, pause and stop self-sufficiently within their VMs. Moreover, one may freely configure each of the VMs and run them through a choice of hardware-based virtualization or software-enabled virtualization, if this is supported by the underlying host hardware. Therein, the guest OS or host OSes and applications are able to interconnect with one another through various mechanisms, which includes universal clipboards and virtualized coverage facilities. Guest VMs can also communicate with one another directly if configured to do so (Oracle Corporation, 2019).



For the purpose of this research, a free version of Oracle Virtualbox Manager was downloaded from the link: <https://www.virtualbox.org/wiki/Downloads>

### 3.4 Raspberry Pi OS

The operating systems of Raspberry Pi are available in different third party images such as Ubuntu, Debian and Windows 10 IoT core. However, the Raspberry Pi Desktop is the primary operating system for PC.

For the purpose of this study, the Debian Stretch with Raspberry Pi Desktop was downloaded from the link: <https://www.raspberrypi.org/downloads/raspberry-pi-desktop/>. Furthermore, after downloading the Debian Raspberry Pi image, the next step was to create the virtual machine on the Oracle Virtualbox, by clicking the 'New' button on the launch tab.

The researcher then proceeded with naming the OS with the desired name, then selecting the OS type and the flavour version of it and the platform architecture such as either 32-bit or 64-bit. The next step required choosing a proper memory size for the operating system, and creating the file size of the OS as well as the type of hard disk file type (i.e., Virtualbox disk image).

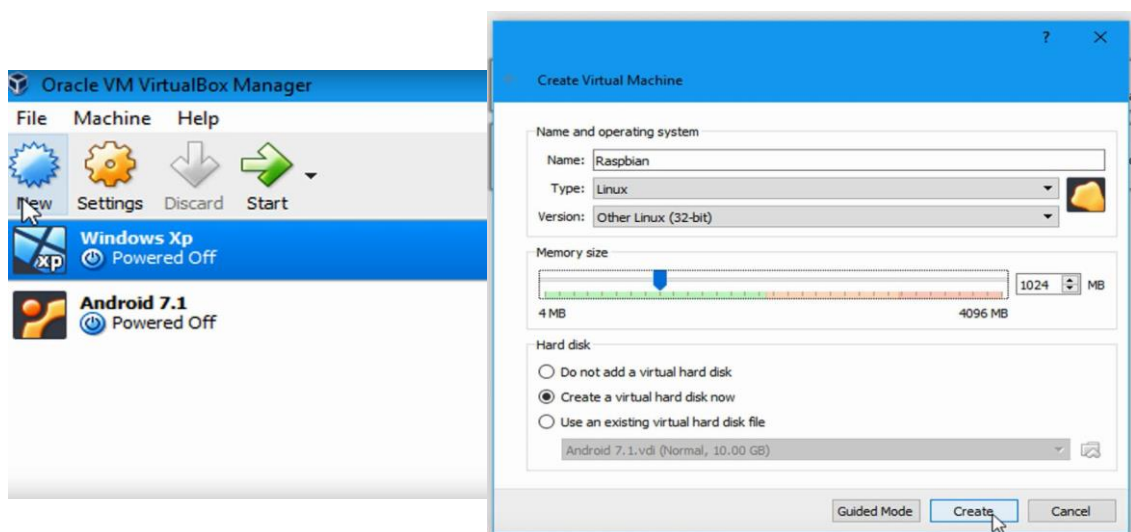


Figure 16. Creating and setting up the new OS on VirtualBox.

The subsequent step involved selecting the location of the creation of OS, the file size of the OS installation files and size of the file, and then selecting the type of the file and standard of storage.

The steps listed below were undertaken by the researcher, which follow the requirement of the OS and software resourced to simulate the same performance of Raspberry Pi 2, from hardware and software point of view:

- i. Selection of the OS setting to modify the storage configuration and booting source for the installation image of Raspberry Pi virtual image (see Figure 17).

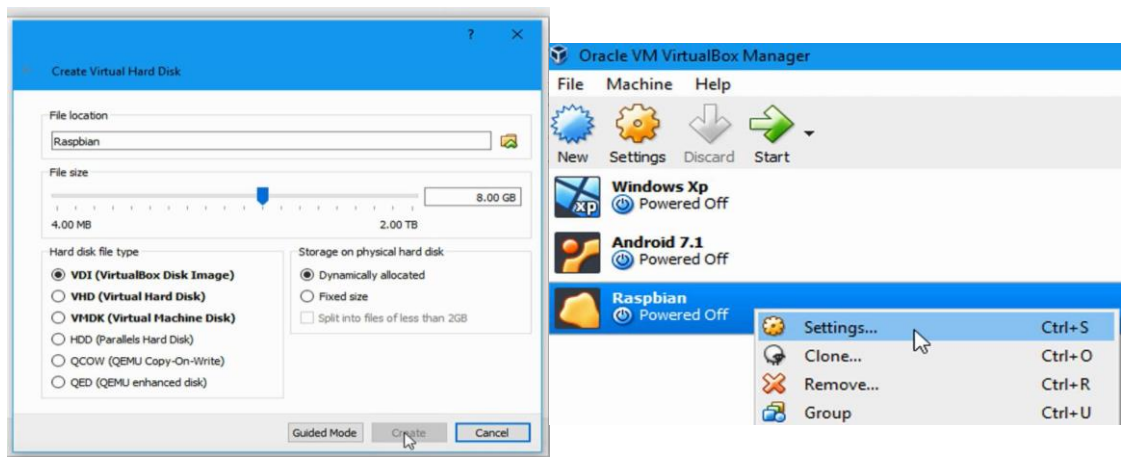


Figure 17. Setup the OS and operation requirement & modify the image setting.

- ii. Configuration of the controller storage of the Integrated Development Environment (IDE), by adding the disk image to locate the Raspberry Pi image that was downloaded; this was done by selecting the disk, exploring the browser to indicate the downloaded raspberry Pi image (see Figure 18).

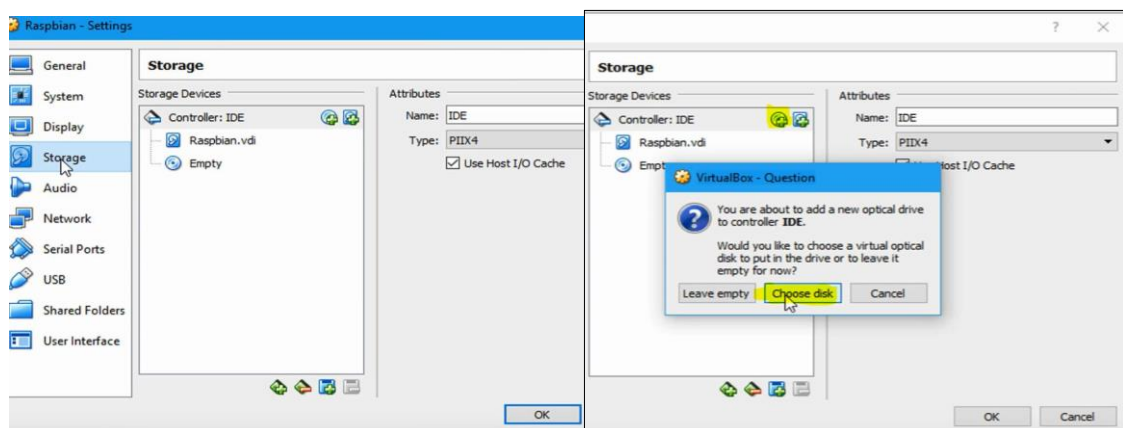


Figure 18. Adding the Raspberry Pi image to the storage.

- iii. Starting of the virtual OS of Raspberry Pi Image and loading the installation process, and therein selecting graphical install from the menu (see Figure 19).
- iv. Loading and copying the system files from Raspberry Pi Image to the selected device storage.

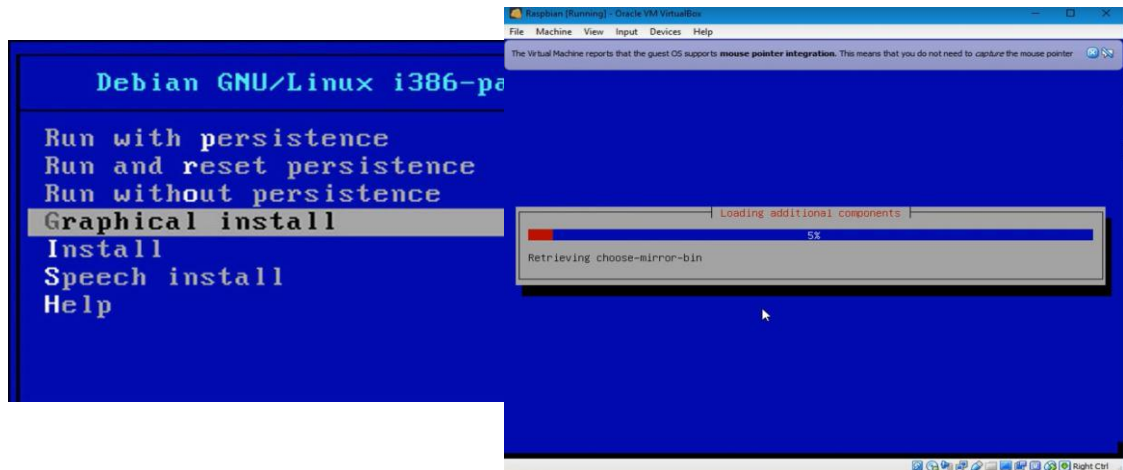


Figure 19. Graphical install process & loading additional components.

- v. Partitioning a disk using different standard schemes, with guided partitioning automation process taking place to partition the entire disk (see Figure 20).
- vi. Selecting a partition table for the SCSI1 (0,0,0) SDA for writing the data on it and allowing changes to the storage media (see Figure 20).

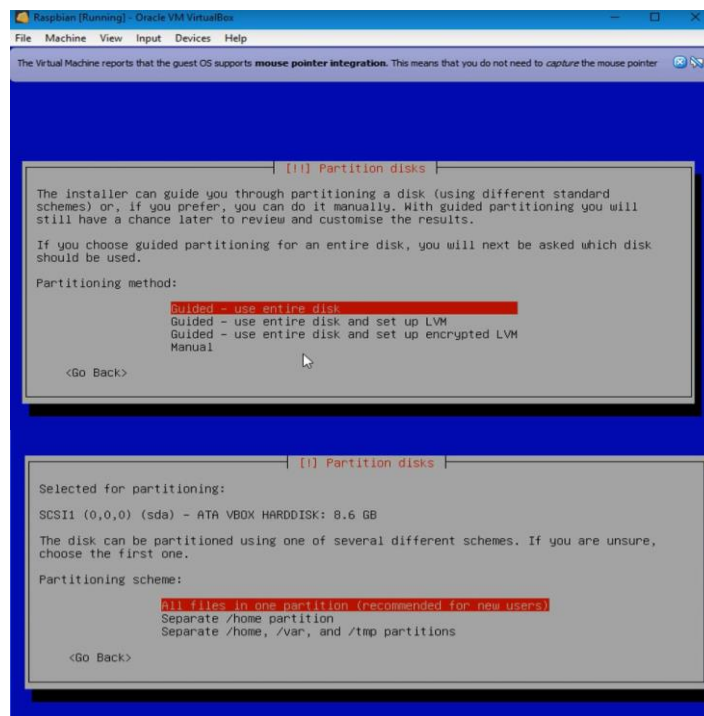


Figure 20. Guided Disk Partitioning.

- vii. Copying data to disk, running console step up /OpenSSH server step up, storing language, installing extra packages – retrieving and installing pciutils, running dpkg, configuring apt that required running services (See Figure 21).
- viii. Installing the GRUB boot loader on a hard disk to the master boot record by selecting the device storage:  
“/dev/sda (ata-Vbox\_HARDDISK\_VBbabd7dae-4ec328a4)”.

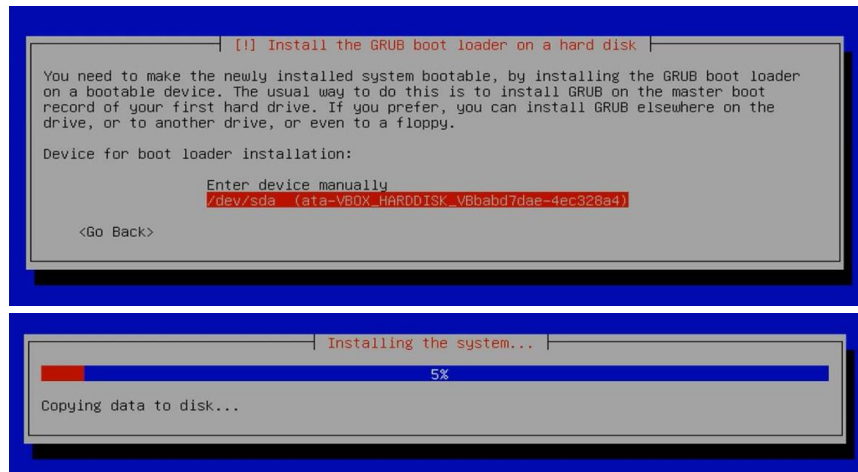


Figure 21. Guided Disk Partitioning, installing services and copying to disk.

- ix. Once the installation was complete, booting into the new system by restarting the system (see Figure 22).

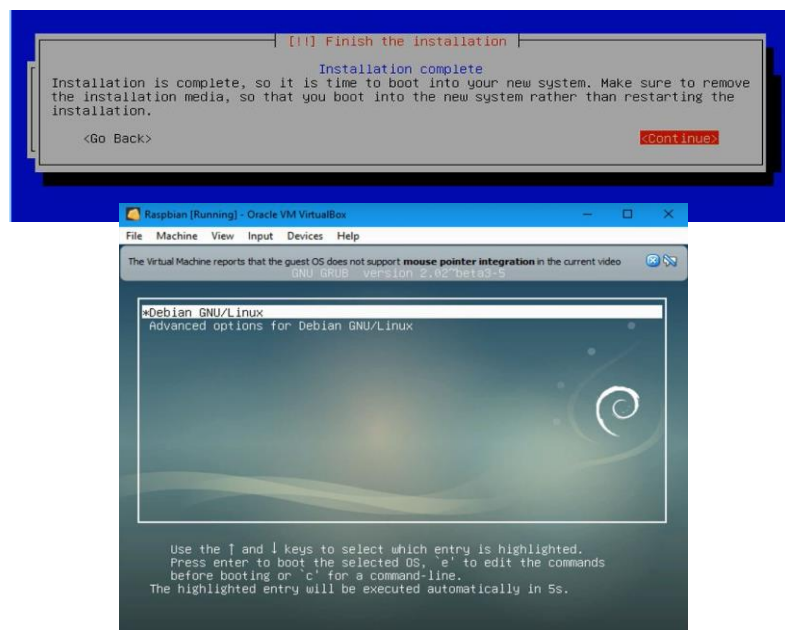


Figure 22. Finish the installation and restart.

```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~$ sudo apt-get update
Hit http://archive.raspberrypi.org jessie InRelease
Hit http://mirrordirector.raspbian.org jessie InRelease
Hit http://archive.raspberrypi.org jessie/main armhf Packages
Hit http://mirrordirector.raspbian.org jessie/main armhf Packages
Hit http://mirrordirector.raspbian.org jessie/contrib armhf Packages
Hit http://archive.raspberrypi.org jessie/ui armhf Packages
Hit http://mirrordirector.raspbian.org jessie/non-free armhf Packages
Hit http://mirrordirector.raspbian.org jessie/rpi armhf Packages
Ign http://archive.raspberrypi.org jessie/main Translation-en_GB
Ign http://archive.raspberrypi.org jessie/ui Translation-en_GB
Ign http://archive.raspberrypi.org jessie/ui Translation-en
Ign http://mirrordirector.raspbian.org jessie/contrib Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/contrib Translation-en
Ign http://mirrordirector.raspbian.org jessie/main Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/non-free Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/non-free Translation-en
Ign http://mirrordirector.raspbian.org jessie/rpi Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/rpi Translation-en
Reading package lists... Done
pi@raspberrypi:~$

pi@raspberrypi: ~
File Edit Tabs Help
Hit http://archive.raspberrypi.org jessie InRelease
Hit http://mirrordirector.raspbian.org jessie InRelease
Hit http://archive.raspberrypi.org jessie/main armhf Packages
Hit http://mirrordirector.raspbian.org jessie/main armhf Packages
Hit http://mirrordirector.raspbian.org jessie/contrib armhf Packages
Hit http://archive.raspberrypi.org jessie/ui armhf Packages
Hit http://mirrordirector.raspbian.org jessie/non-free armhf Packages
Hit http://mirrordirector.raspbian.org jessie/rpi armhf Packages
Ign http://archive.raspberrypi.org jessie/main Translation-en_GB
Ign http://archive.raspberrypi.org jessie/ui Translation-en_GB
Ign http://archive.raspberrypi.org jessie/ui Translation-en
Ign http://mirrordirector.raspbian.org jessie/contrib Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/contrib Translation-en
Ign http://mirrordirector.raspbian.org jessie/main Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/main Translation-en
Ign http://mirrordirector.raspbian.org jessie/non-free Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/non-free Translation-en
Ign http://mirrordirector.raspbian.org jessie/rpi Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/rpi Translation-en
Reading package lists... Done
pi@raspberrypi:~$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree... 50%

```

Figure 23. Updating and upgrading the Raspberry Pi Package.

After completing the above steps, Raspberry Pi Desktop starts the operating system requirements to an updated version of packages which can be obtained by executing ***sudo apt-get update*** (see Figure 23); before any package installation, it is recommended to install the update and is also essential to run it for installing the latest updates, even if one has not removed or added any Software Sources.

### 3.5 Antivirus & Firewall

ClamAV is an open-source anti-virus which is compatible with a number of operating systems. In other words, no subscriptions or license is required to update the antivirus signature to the latest and updated database. Moreover, its detection rate is considered to be good among users, with its working speed being light on CPU utilization as well. Furthermore, ClamAV comprises of command-line utilities for on-demand file scanning, a multi-threaded scanner daemon and automatic signature updates. Nonetheless, for installing this open-source antivirus, the following command was executed (Figure 24):

```
# Sudo apt-get install ClamAV
```



Once the installation was complete, ClamAV signatures were updated, and a freshclam command was run by entering the following command, in order to update the signatures database manually (see Figure 24):

```
# Sudo freshclam
```

The figure shows two terminal windows side-by-side. The left window shows the process of adding a user named 'hossam' and installing ClamAV. The right window shows the execution of the 'sudo freshclam' command.

```

pi@raspberrypi: ~
File Edit Tabs Help
via the NAME_REGEX configuration variable. Use the '--force-badname'
option to relax this check or reconfigure NAME_REGEX.
root@raspberrypi:/home/pi# sudo adduser hossam
Adding user 'hossam' ...
Adding new group 'hossam' (1001) ...
Adding new user 'hossam' (1001) with group 'hossam' ...
Creating home directory '/home/hossam' ...
Copying files from '/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for hossam
Enter the new value, or press ENTER for the default
  Full Name []: Hossam
  Room Number []: kher
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
root@raspberrypi:/home/pi#
root@raspberrypi:/home/pi#
root@raspberrypi:/home/pi#
root@raspberrypi:/home/pi# sudo apt-get install clamav

pi@raspberrypi: ~
File Edit Tabs Help
root@raspberrypi:~#
root@raspberrypi:~# sudo freshclam

```

Figure 24. Installing ClamAV Antivirus and updating the signature DB.

The figure shows a terminal window with the output of the 'clamscan -r --bell -i /home/bill/Downloads' command. The output indicates that no files were scanned and no viruses were detected.

```

pi@raspberrypi: ~
File Edit Tabs Help
root@raspberrypi:~#
root@raspberrypi:~# clamscan -r --bell -i /home/bill/Downloads
/home/bill/Downloads: No such file or directory
WARNING: /home/bill/Downloads: Can't access file

----- SCAN SUMMARY -----
Known viruses: 6124090
Engine version: 0.100.3
Scanned directories: 0
Scanned files: 0
Infected files: 0
Data scanned: 0.00 MB
Data read: 0.00 MB (ratio 0.00:1)
Time: 25.530 sec (0 m 25 s)
root@raspberrypi:~#

```

Figure 25. ClamAV scanning the Operating System.

Once the antivirus was installed and database signatures were updated to the latest virus database, virus scan was run to scan the entire operating system; the sound bell function was also used to alert the researcher once the virus is detected, by executing the following command (see Figure 25):

```
# clamscan -r --bell -i /home/
```

Securing the operating system with a firewall is mandatory to protect the internal network from external attackers and intruders. For this purpose, an Uncomplicated Firewall (UFW) can be used. UFW is an operator-friendly front-end firewall that can manage iptable firewall guidelines; its key aim is to manage tables easily. Thus, the next step involved installing the package UFW (see Figure 26) by executing the command:

```
# sudo apt install ufw
```

```

hossam@raspberrypi: ~
File Edit Tabs Help
root@raspberrypi:~# sudo apt install ufw
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  ufw
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 164 kB of archives.
After this operation, 848 kB of additional disk space will be used.
Get:1 http://ftp.debian.org/debian stretch/main 1386 ufw all 0.35-4 [164 kB]
Fetched 164 kB in 1s (137 kB/s)
Preconfiguring packages ...
Selecting previously unselected package ufw.
(Reading database ... 70%
hossam@raspberrypi: ~
File Edit Tabs Help
Preconfiguring packages ...
Selecting previously unselected package ufw.
(Reading database ... 124614 files and directories currently installed.)
Preparing to unpack ../archives/ufw_0.35-4_all.deb ...
Unpacking ufw (0.35-4) ...
Setting up ufw (0.35-4) ...

Creating config file /etc/ufw/before.rules with new version
Creating config file /etc/ufw/before6.rules with new version
Creating config file /etc/ufw/after.rules with new version
Creating config file /etc/ufw/after6.rules with new version
Created symlink /etc/systemd/system/multi-user.target.wants/ufw.service → /lib/s
ystemd/system/ufw.service.
Processing triggers for systemd (232-25+deb9u11) ...
Processing triggers for man-db (2.7.6.1-2) ...
Processing triggers for rsyslog (8.24.0-1) ...
root@raspberrypi:~# sudo ufw enable
Firewall is active and enabled on system startup
root@raspberrypi:~# sudo ufw status
Status: active
root@raspberrypi:~#

```

Figure 26. Installing UFW Firewall.

After the installation of the UFW package on the Raspberry Pi OS, status verification was conducted by executing the following command:

```
# sudo UFW status
```

### 3.6 RootKit & Malware Scanner

Rootkit Hunter (rkhunter) is an open-source Linux/Unix based scanning device for Linux systems. This scanner was released under the General Public Licence (GPL) for scanning backdoors, local exploits and rootkits on a system. As such, it helps in scanning wrong authorizations set on binaries, unknown files, malicious strings in the kernel, etc. Since rkhunter is found in the Extra Packages for Enterprise Linux (EPEL) repository, installing them was straight forward (see Figure 27) by executing the command line:

```
#sudo apt-get install rkhunter
```

```

root@raspberrypi:~# sudo apt-get install rkhunter
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bsd-mailx exim4-base exim4-config exim4-daemon-light lsof unhide unhide.rb
Suggested packages:
  eximon4 exim4-doc-html | exim4-doc-info spf-tools-perl swaks
The following NEW packages will be installed:
  bsd-mailx exim4-base exim4-config exim4-daemon-light lsof rkhunter unhide unhide.rb
0 upgraded, 8 newly installed, 0 to remove and 68 not upgraded.
Need to get 2,736 kB of archives.
After this operation, 5,733 kB of additional disk space will be used.
Do you want to continue? [Y/n] █

```

Figure 27. Installing rkhunter Rootkit package.

In the next step, the rkhunter RootKit scanner was run to check all the services that are installed by default, especially the login method to the operating system such as Secure Shell (SSH) protocol (see Figure 28). For instance, once the rkhunter check is started, SSH login should be confirmed as secure such that the attacker root user is unable to log in via SSH. As such, only a non-privileged account must be used to log in. Thus, in this step, the researcher made use of sudo once access had been gained.

```

root@raspberrypi:~# sudo rkhunter -c --enable all --disable none
[ Rootkit Hunter version 1.4.2 ]

Checking system commands...

Performing 'strings' command checks

Performing file properties checks
Checking for prerequisites
/usr/sbin/adduser [ OK ]
/usr/sbin/chroot [ OK ]
/usr/sbin/cron [ OK ]
/usr/sbin/groupadd [ OK ]
/usr/sbin/groupdel [ OK ]
/usr/sbin/groupmod [ OK ]
/usr/sbin/grepck [ OK ]
/usr/sbin/nologin [ OK ]
/usr/sbin/pwck [ OK ]
/usr/sbin/rsyslogd [ OK ]
/usr/sbin/sshd [ OK ]
/usr/sbin/tcpd [ OK ]
/usr/sbin/useradd [ OK ]
/usr/sbin/userdel [ OK ]
/usr/sbin/usermod [ OK ]
/usr/sbin/vipw [ OK ]
/usr/sbin/unhide [ OK ]
/usr/sbin/unhide-linux [ OK ]
/usr/sbin/unhide-posix [ OK ]
/usr/sbin/unhide-tcp [ OK ]
/usr/bin/awk [ OK ]
/usr/bin/basename [ OK ]
/usr/bin/chatr [ OK ]
/usr/bin/curl [ OK ]
/usr/bin/cut [ OK ]
/usr/bin/diff [ OK ]
/usr/bin/dirname [ OK ]
/usr/bin/dpkg [ OK ]
/usr/bin/dpkg-query [ OK ]
/usr/bin/du [ OK ]

```

Figure 28. Performing rkhunter scanning.

```

Checking application versions...
Checking version of Exim MTA [ OK ]
Checking version of GnuPG [ OK ]
Checking version of OpenSSL [ OK ]
Checking version of OpenSSH [ OK ]

System checks summary
=====
File properties checks...
Files checked: 146
Suspect files: 1

Rootkit checks...
Rootkits checked : 377
Possible rootkits: 0

Applications checks...
Applications checked: 4
Suspect applications: 0

The system checks took: 2 minutes and 29 seconds

All results have been written to the log file: /var/log/rkhunter.log

One or more warnings have been found while checking the system.
Please check the log file (/var/log/rkhunter.log)

root@raspberrypi:~#

```

Figure 29. Rkhunter system checks summary.

Subsequently, rkhunter scanning was performed for all suspected files and applications against rootkits and malware (see Figure 29); the scanning included files and directories as well as TCP/IP services running on the operating system. It must be noted that there are two types of rkhunter scanning: manual scans and automatic. The major difference



between the two is that the speed varies, i.e., the manual method is slower than the automatic scan due to the requirement to show the output on the screen and write to the log, after the system check is finished.

### 3.7 Intrusion Detection & Prevention System

BriarIDS is a home intrusion detection system (IDS) solution for Raspberry Pi. The name originates from the protection given to rabbits by Briars and Brambles (Raspberry bushes) when under attack. Nonetheless, BriarIDS offers a simple and affordable solution that can equip home users with their very own personal/home network IDS using the Raspberry Pi unit. This is because BriarIDS is a combination of two worlds of security extensive layers and defense mechanisms, as it provides IPS as well as IDS features. Furthermore, BriarIDS consists of many options in one package image; the features that are presented in the image are listed below:

- i. Bro with Critical Stack intel feed
- ii. Suricata Intrusion Detection and Prevention

From the above, Bro is an open networking security platform source that illuminates networking activities in detail, which can be positioned at scale. Thus, it helps in providing a complete platform for more regular web traffic analysis, and it includes security features like attack detection, event correlation and log recording. Moreover, Bro has its own signature-based tools to allow rapid tracking of logs that are generated once events across different traffic flow.

The second feature listed above, i.e., Suricata, is an IDS and IPS engine introduced in 2009, in an effort to meet the modern infrastructural demands. Suricata, just like Snort, is rule-based and is compatible with Snort Rules. Additionally, multi-threading has been introduced in this, which offers the theoretical capability to process additional rules across speedy networks, with higher traffic volumes through the same hardware. For instance, Rubens (2015) argues: "Since it happens to be multi-threaded, one application will help in balancing the processing load across each processor on the sensor where Suricata is configured, enabling commodity hardware to attain 10-gigabyte speeds without compromising ruleset coverage."

Python Pip modules are one of the main requirements to install BriarIDS packages. As such, Pip happens to be a package management system that simplifies installation and

manages software packages written in Python, such as those found in the Python Package Index. Additionally, the missing security sector is the IDS for the Raspberry Pi. BriarIDS is configured to function with the Raspbian operating system and takes benefits of the PyQT Graphical User Interface (GUI) frontend security layer for an all-in-one solution that monitors home networks. In addition, Bro is integrated into the BriarIDS GUI for additional logging options as well as Suricata, as part of the solution appliance (Robbie, 2013). The instruction to install BriarIDS on the Raspbian OS and configure both Bro and Suricata was by following the specific sequence as below (see also Figure 30):

- i. First, install pip: `sudo apt-get install python-pip` (should be on raspi by default)
- ii. Next, clone the repo! `git clone https://github.com/musicmancorley/BriarIDS.git`
- iii. Then, `cd` into the directory: `cd BriarIDS`
- iv. Finally, `sudo python setup.py install`

```

pi@raspberrypi: ~
File Edit Tabs Help
root@raspberrypi:~# sudo apt-get install python-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
python-pip is already the newest version (9.0.1-2+rpt2).
0 upgraded, 0 newly installed, 0 to remove and 68 not upgraded.
root@raspberrypi:~# git clone https://github.com/musicmancorley/BriarIDS.git
Cloning into 'BriarIDS'...
remote: Enumerating objects: 692, done.
remote: Total 692 (delta 0), reused 0 (delta 0), pack-reused 692
Receiving objects: 100% (692/692), 13.11 MiB | 258.00 KiB/s, done.
Resolving deltas: 100% (408/408), done.
root@raspberrypi:~# cd BriarIDS
root@raspberrypi:~/BriarIDS#
pi@raspberrypi: ~
File Edit Tabs Help
root@raspberrypi:~/BriarIDS# briar
pyqt5 not installed..installing now.
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libqt5clucene5 libqt5designer5 libqt5help5 libqt5test5 python-sip
Suggested packages:
  python-pyqt5-dbg
The following NEW packages will be installed:
  libqt5clucene5 libqt5designer5 libqt5help5 libqt5test5 python-pyqt5 python-sip
0 upgraded, 6 newly installed, 0 to remove and 68 not upgraded.
Need to get 5,471 kB of archives.
After this operation, 22.4 MB of additional disk space will be used.
Get:1 http://ftp.debian.org/debian stretch/main i386 libqt5clucene5 i386 5.7.1-1 [294 kB]
Get:2 http://ftp.debian.org/debian stretch/main i386 libqt5designer5 i386 5.7.1-1 [2,888 kB]
31% [2 libqt5designer5 1,612 kB/2,888 kB 56%]

```

Figure 30. Installing BriarIDS GUI.

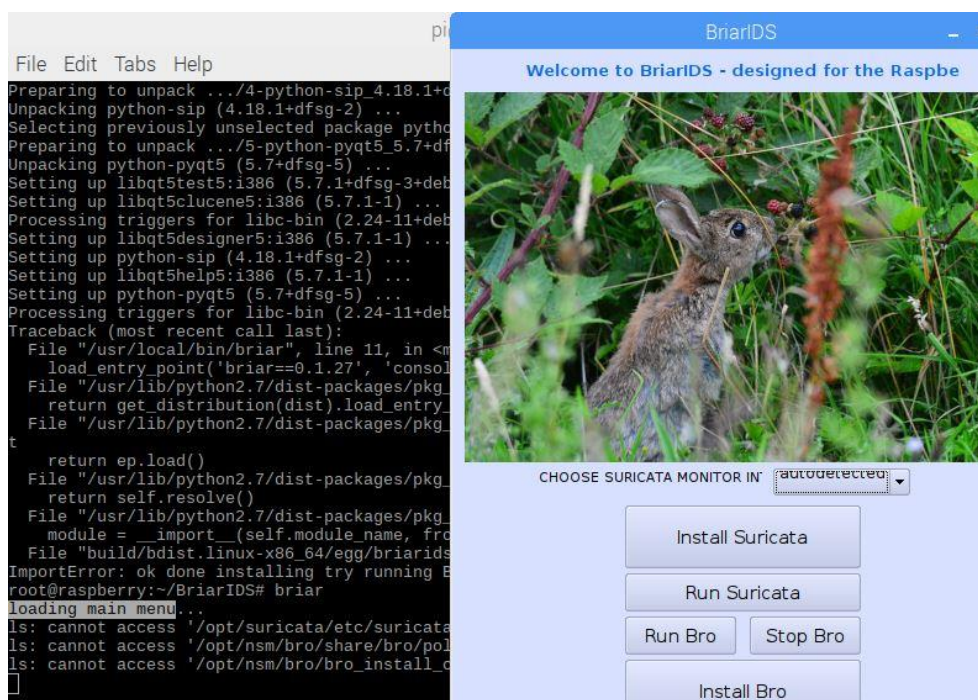


Figure 31. Installing BriarIDS GUI.

The installation therein started building a dependency tree and read stated information for the installation package (pyqt5); after this operation is completed, the GUI interface menu was loaded (see Figure 31).

### 3.8 Network-Wide Ad Blocking

The Pi-Hole is used as a DNS sinkhole, which safeguards one's devices from useless content without the installation of any client-side software. Its advantages are as follows:

- Easy-to-install: the versatile installer runs through the process, and takes less than 10 minutes.
- Resolute: contents are blocked in non-browser places, such as ad-laden mobile applications and smart Televisions.
- Responsive: seamlessly makes everyday browsing faster by caching DNS queries.
- Lightweight: plays smoothly with minimal software and hardware requirements.
- Robust: a command-line platform with quality assurance for interoperability.
- Insightful: an outstanding responsive Internet interface dashboard for viewing and controlling user Pi-Hole.
- Versatile: functions optionally like a DHCP server that ensures all devices are automatically protected.

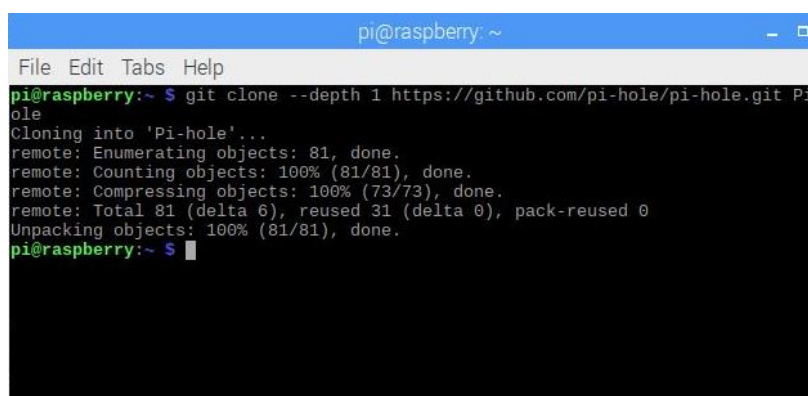
- Scalable: capacity to handle millions of queries during installation on server-grade hardware.
- Modern: blocking of ads on both Internet Protocol version 6 (IPv6) and Internet Protocol version 4 (IPv4).

Pi-Hole also happens to be a Linux network-level Internet ad-tracker blocking app, acting like a DNS sinkhole (or a DHCP server optionally), intended for use on private networks. It is manufactured to be used on an embedded device with network capabilities like the Raspberry Pi, but it is also possible to use it on other systems running cloud and Linux implementations. Generally, Pi-Hole can block traditional web advertisements and also advertises in unconventional things, such as cellular operating systems and smart TVs (Catchpoint, 2015).

With the DNS Sinkhole functionality of the Pi-Hole, it can be used as a DNS server on a network, and as such, it will respond with a fake address when a client software or browser attempts to load content from a known advertiser. Although Pi-Hole maintains a set of 'blocklists,' nonetheless, for the purpose of this research, these are further supplemented with additional blocklists (Crocker-White, 2019),

To setup Pi-Hole from the command prompt, the below command was used in sequence, as shown in Figure 32:

```
# git clone --depth 1 https://github.com/pi-hole/pi-hole.git Pi-hole
```



```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~$ git clone --depth 1 https://github.com/pi-hole/pi-hole.git Pi-hole
Cloning into 'Pi-hole'...
remote: Enumerating objects: 81, done.
remote: Counting objects: 100% (81/81), done.
remote: Compressing objects: 100% (73/73), done.
remote: Total 81 (delta 6), reused 31 (delta 0), pack-reused 0
Unpacking objects: 100% (81/81), done.
pi@raspberrypi:~$

```

Figure 32. Cloning into Pi-Hole.

The Pi-Hole GUI installation starts with a welcome page and confirmation of transforming the Raspberry Pi device into a Network-Wide adblocker (see Figure 33).

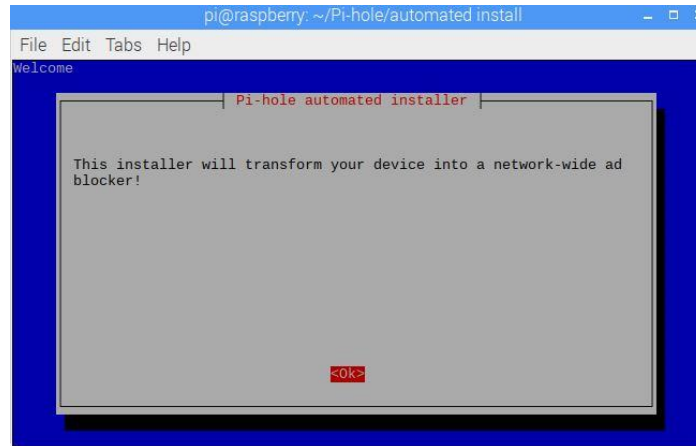


Figure 33. Raspberry Pi transformed into network-wide adblocker.

However, the Pi-Hole happens to be a server that requires a static IP address for functioning smoothly (see Figure 34). As such, one can choose to use present network settings (DHCP), or they can edit them manually, as discussed further below. Nonetheless, the step-by-step procedure undertaken for this research are elaborated hereunder:

- i. Upstream DNS provider was chosen for the purpose of this research (see Figure 34). In order to use one's own, the user can select a custom option from the service providers list such as Google, OpenDNS, Level13, Comodo, DNSwatch, Quad9 etc.

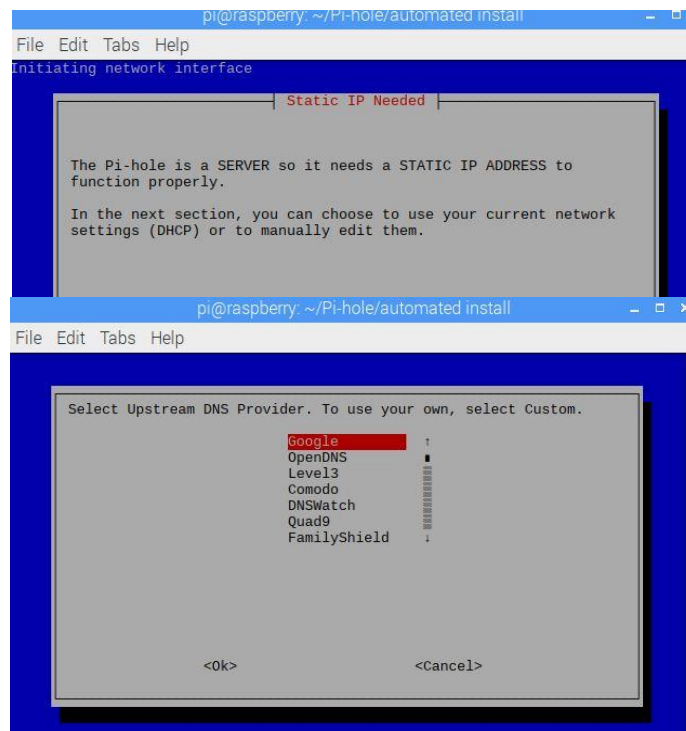


Figure 34. Setting up Pi-Hole.

As the Pi-Hole counts on a 3rd-party list for blocking ads, one may make use of the suggestion used in this research, and/or add one's own post installation. For deselecting any list, the user can make use of the spacebar and arrow key (see Figure 35).

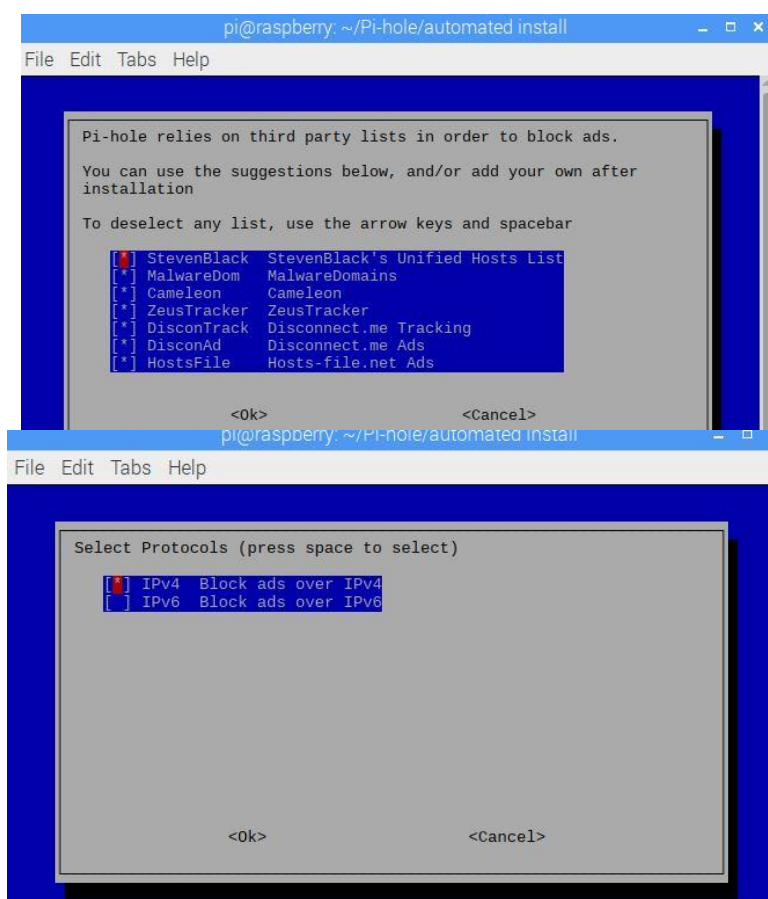


Figure 35. Pi-Hole Third Party Services and TCP/IP protocols.

- ii. With regard to the static IP address required by the Pi-Hole for functioning properly, the IPv4 protocol was selected here, as shown in Figure 35 (a DHCP reservation will work adequately). There is a possibility of users running into issues because this experiment had installed `dhcpcd5`, which may conflict with other running network managers such as `client`, `dhcpcd`, `Network Manager`, and `system-network`.
- iii. Cross-checking the network and static IP details was the next step. Gateway IP shown in Figure 36 was the router's IP address, which Pi-Hole installer needed to pull automatically.



- iv. Additionally, assigning local IP address was required, and hence it was entered as 192.168.28.158/24 and gateway was set as 192.168.28.2 (see Figure 36).
- v. The next step involved enabling the Web Admin interface (see Figure 36).

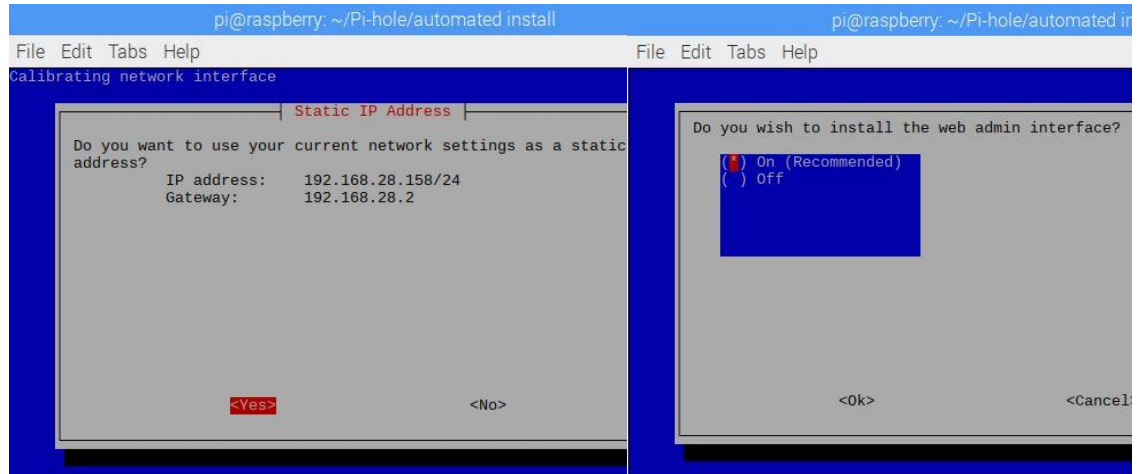


Figure 36. Configure management IP address and install web admin interface.

To block the advertisement from suspicious domains on a network level, including every device linked to the home coverage, as per the configured third-party DNS independency is on the Google DNS server.

The Pi-Hole installation package then copies the files to the installation directory (see Figure 37), with the customized setting of network address and feature (Google DNS, DHCP table lookup). Furthermore, Pi-Hole depends on upstream and DNS queries log to filter and block malicious domains, exploit kit domains, adware, malware and phishing domains.

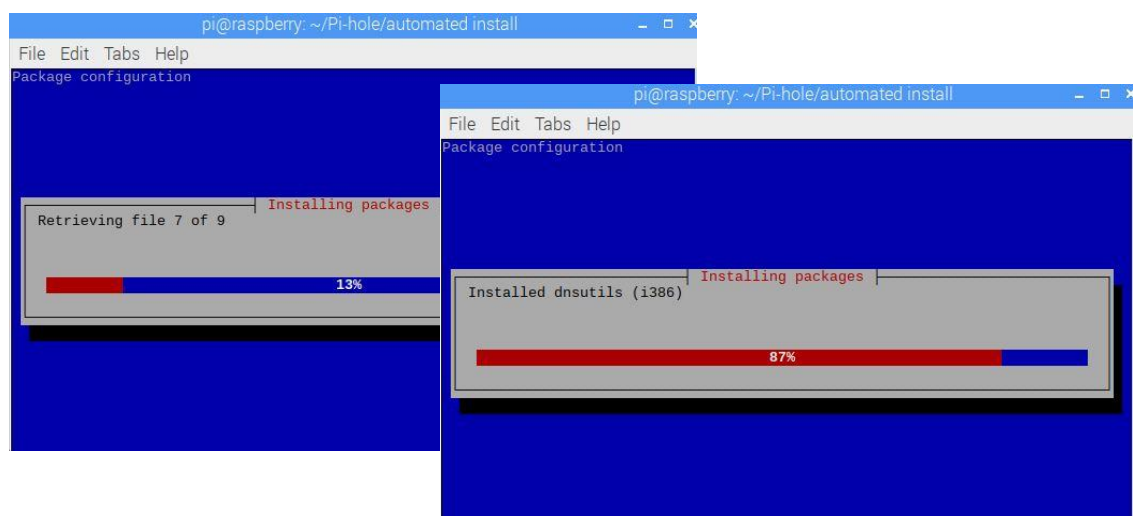
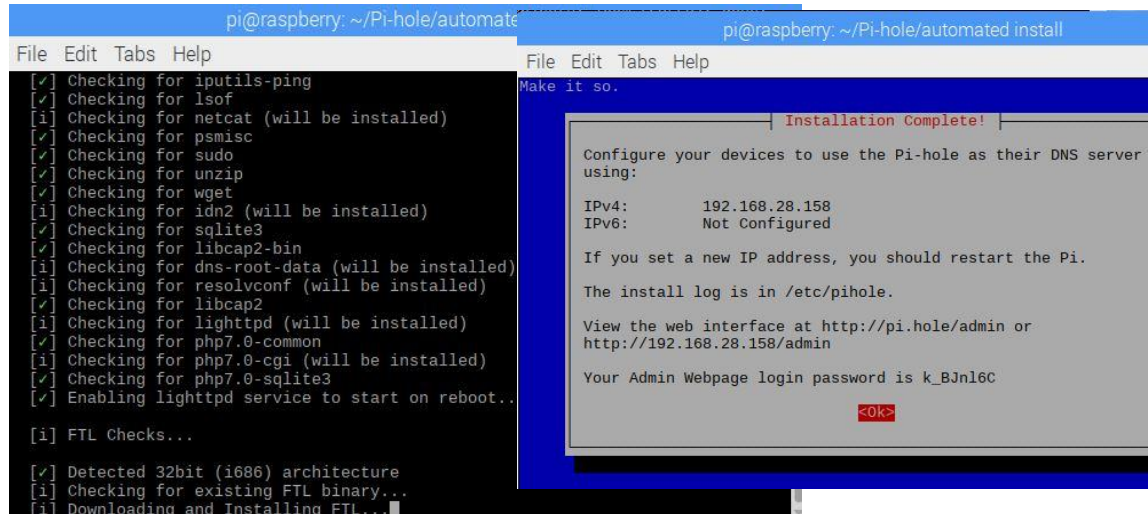


Figure 37. Pi-Hole installation process.

The installer screen then showed the password to use for the web interface (the reader may refer to the commands listed later in this guide). Thus, all steps to install and configure Pi-Hole were now complete (see Figure 38).



```

pi@raspberrypi: ~/Pi-hole/automate
File Edit Tabs Help
[✓] Checking for iputils-ping
[✓] Checking for lsof
[i] Checking for netcat (will be installed)
[✓] Checking for psmisc
[✓] Checking for sudo
[✓] Checking for unzip
[✓] Checking for wget
[i] Checking for idn2 (will be installed)
[✓] Checking for sqllite3
[✓] Checking for libcap2-bin
[i] Checking for dns-root-data (will be installed)
[i] Checking for resolvconf (will be installed)
[✓] Checking for libcap2
[i] Checking for lighttpd (will be installed)
[✓] Checking for php7.0-common
[i] Checking for php7.0-cgi (will be installed)
[✓] Checking for php7.0-sqlite3
[✓] Enabling lighttpd service to start on reboot...

[i] FTL Checks...

[✓] Detected 32bit (i686) architecture
[i] Checking for existing FTL binary...
[i] Downloading and Installing FTL...

pi@raspberrypi: ~/Pi-hole/automated install
File Edit Tabs Help
Make it so.

Installation Complete!

Configure your devices to use the Pi-hole as their DNS server
using:

IPv4:      192.168.28.158
IPv6:      Not Configured

If you set a new IP address, you should restart the Pi.

The install log is in /etc/pihole.

View the web interface at http://pi.hole/admin or
http://192.168.28.158/admin

Your Admin Webpage login password is k_BJn16C

<Ok>

```

Figure 38. Pi-Hole setup complete.

Additionally, once the installation was complete, a summary of the configurations, including the IP address, URL and the credentials, were shown on the screen for accessing the Pi-Hole Dashboard interface.

As seen in Figure 39 below, the Pi-Hole interface consists of many sections, such as:

- Status
- Queries Block
- Percent blocked
- Domain on blocklist.

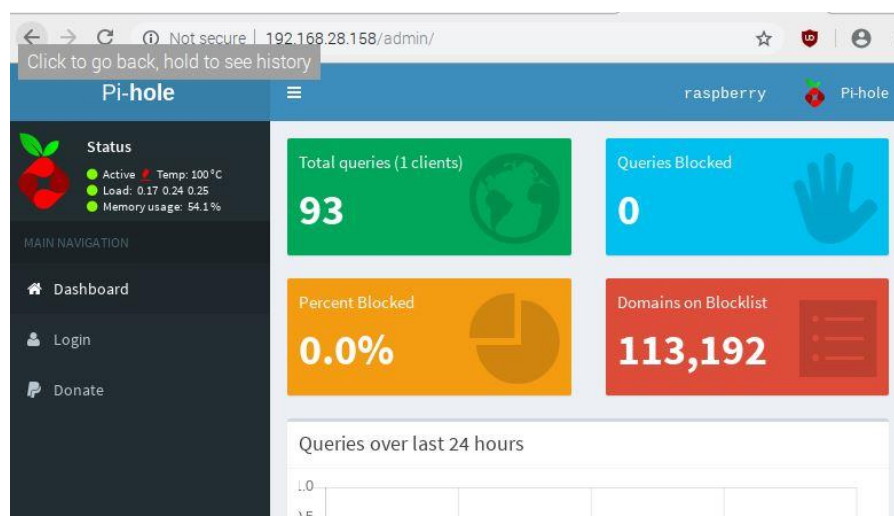


Figure 39. Pi-Hole web Dashboard.



Updating the Pi-Hole Database and engine version is required to improve its accuracy and performance, which was the next step undertaken in this study (see Figure 40).

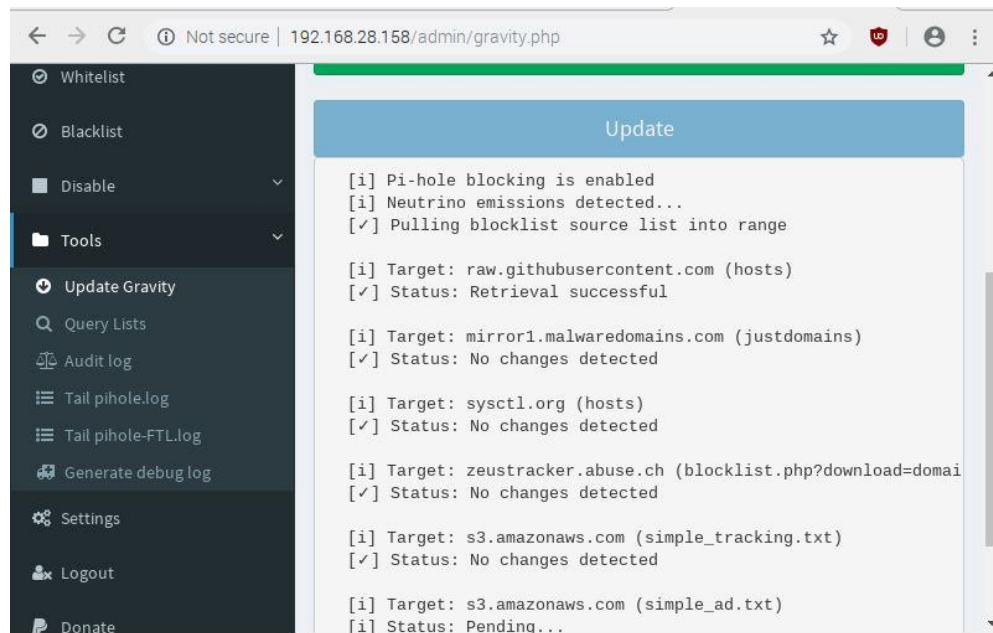


Figure 40. Pi-Hole Database and signature updates.

## 4 Post-Implementation Analysis

For the purpose of validating the research outcome, security measurement tests were performed by simulating different malicious attacks. The purpose of these tests was to ensure the functionality and accuracy of the inline security measures used in this Internet home security solution. Thus, in this chapter, the snapshots from each exercise are explained with an appropriate description.

### 4.1 Intrusion Prevention & Detection Tests

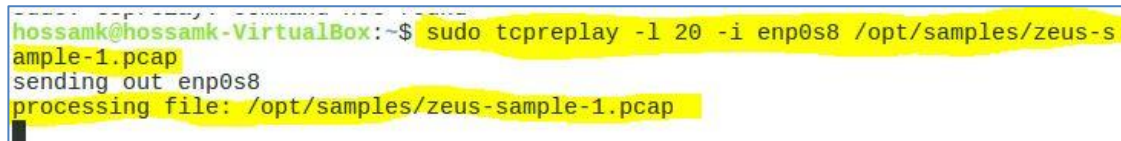
The following scenario simulates the intrusion prevention system's functionality and also alerts the user with a system notification when there is an attack. In order to accomplish this, a packet replay technique was used to run malicious traffic targeting the victim host (i.e. Internet Home Security). Thus, using TCP replay, the packet is replayed as if the attack is occurring live over the wire.

To locate the traffic attack sample, the following is keyed in:

```
# Locate Zeus
```

To execute the TCP replay command, the following syntax is entered (see Figure 41):

```
# sudo tcpreplay -l 20 -i enp0s8 -t /opt/samples/zeus-sample-1.pcap
```



```
hossamk@hossamk-VirtualBox:~$ sudo tcpreplay -l 20 -i enp0s8 /opt/samples/zeus-sample-1.pcap
sending out enp0s8
processing file: /opt/samples/zeus-sample-1.pcap
```

Figure 41. Executing the TCP replay for malicious traffic.

Breakdown of the command syntax:

(20) refers to the number of command loop which is 20 times;

(-i) refers to informational logs;

(enp0s8) refers to the host-only interface;

(t) refers to top speed;

and finally, the full path to Wireshark captured traffic sample is added.

Stanislac and Beardsley (2017) state that Squert is a web application used to query and view event data stored in a Sguil database (typically IDS alert data). Moreover, Squert is a visual tool that attempts to provide additional context to events through the use of metadata, time series representations, and weighted and logically grouped result sets.

Hence, Squert alert monitoring application is used here, which required to be accessed using the preconfigured username and password; on the dashboard of Squert (as seen in Figure 42), the recent alert is shown with the time stamp.

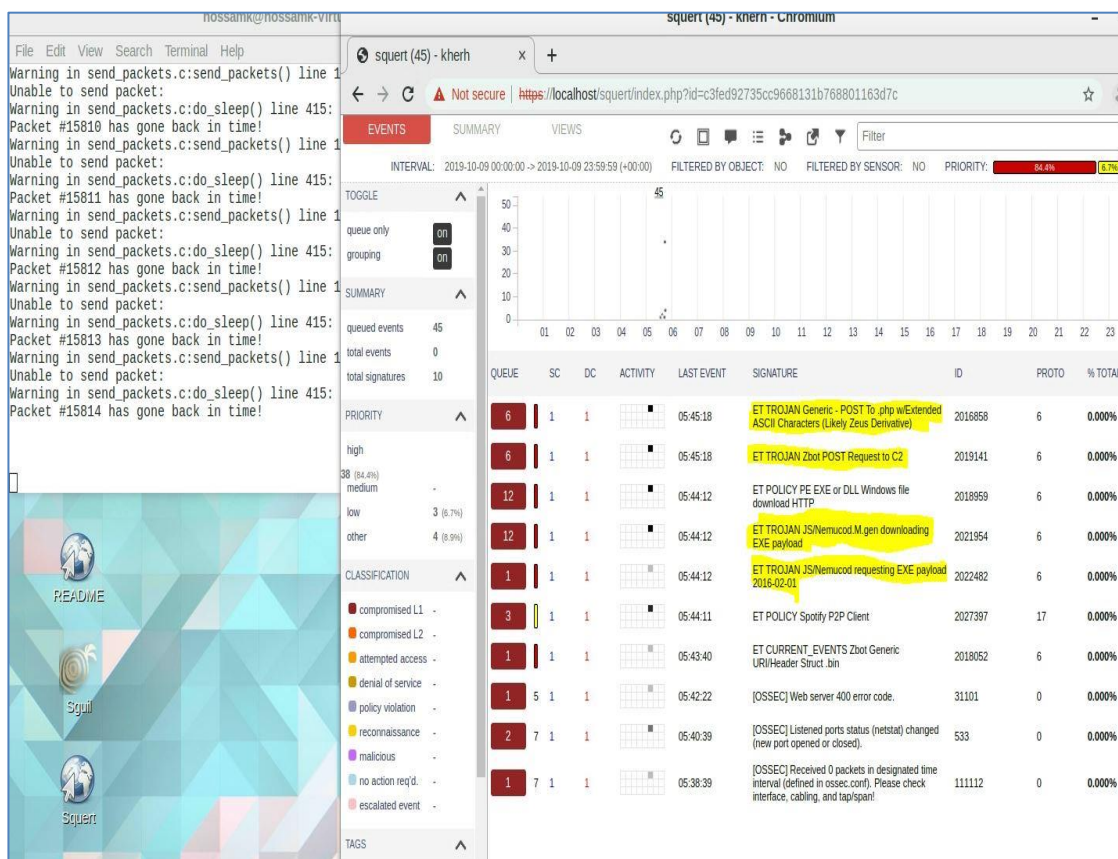


Figure 42. The Squert Alerting Dashboard.

The TCP replay establishes reverse TCP shell to the localhost; Squert then detects the existence of a Trojan exploitation attempt and malicious malware Zbot (see Figure 42).

Breakdown of the alert 'ET Trojan':

- It first appears that an abnormal HTTP GET Request attempted to establish a connection with 188.124.5.100 but did not succeed (see Figure 43); this gets classified as Trojan infection. Although no association was established, this type of incident would still need to be reported because proper action needs to be taken to wipe or isolate the infected packet.
- The incidents alerting dashboard provides visualization of the alerts and classification for all types of alerts; in this particular event, Squert auto categorized it as compromised L1.

QUEUE	SC	DC	ACTIVITY	LAST EVENT	SIGNATURE	ID	PROTO	% TOTAL
10	1	1		05:45:18	ET TROJAN Generic - POST To .php w/Extended ASCII Characters (Likely Zeus Derivative)	2016858	6	0.000%
<p>alert tcp \$HOME_NET any -&gt; \$EXTERNAL_NET \$HTTP_PORTS (msg:"ET TROJAN Generic - POST To .php w/Extended ASCII Characters (Likely Zeus Derivative)"; flow:established,to_server; content:"POST"; http_method; content:".php"; http_uri; content:"Referer[3a]"; http_header; content:"Content-Type[3a]"; http_header; content:" MSIE "; http_header; pcre:"/^\.{0,3}[\x80-\xff]{1,3}[\x00-\x7f]{1,3}[\x80-\xff]{1,3}[\x80-\xff]{1,3}/P"; classtype:trojan-activity; sid:2016858; rev:7; metadata:created_at 2013_05_15, updated_at 2013_05_15;)</p> <p>file: downloaded.rules:16126</p> <p><input checked="" type="checkbox"/> CATEGORIZE 10 EVENT(S) <input type="checkbox"/> CREATE FILTER: <a href="#">src</a> <a href="#">dst</a> <a href="#">both</a></p>								
QUEUE	ACTIVITY	LAST EVENT	SOURCE	AGE	COUNTRY	DESTINATION	AGE	COUNTRY
10		2019-10-09 05:46:25	192.168.3.35	0	RFC1918 (lo)	188.124.5.100	-	unknown (-)

Figure 43. Signature information and Source/destination address.

According to the Suricata Rules document, a rule or signature would contain the following components (Check Point, 2018):

- i. The action, which determines what occurs when the signature matches
- ii. The header, which defines the protocol, IP addresses, ports as well as direction of the rule.
- iii. The rule options, which defines the specifics of the rule.

For example, the TCP stream, when viewed as a rule, is as follows:

```

alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS (msg:"ET
TROJAN Generic - POST To .php w/Extended ASCII Characters (Likely Zeus
Derivative)"; flow:established,to_server; content:"POST"; http_method; con
tent:".php"; http_uri; content:"Referer[3a]"; http_header; content:"Content
Type[3a]"; http_header; content:" MSIE "; http_header; pcre:"/^\.{0,3}[\x80
\xff]{1,3}[\x00-\x7f]{1,3}[\x80-\xff]{1,3}[\x80-\xff]{1,3}/P"; classtype:trojan-activity;
sid:2016858; rev:7

```

In the above traffic sample, the **red component** is the action, **orange components** denote the header, and **blue components** are the options. Thus, in this example, the action is to **alert**; this means that if a signature matches and the rule contains 'alert', the packet will be treated like any other non-threatening packet, however for this particular rule, an alert will be generated by Suricata, which only the system administrator will be notified of.

Therein, Inline/IPS will be able to block the network traffic in two ways – one is by 'drop' and the other by the 'reject' option. Nonetheless, the default order is: pass, drop, reject, alert (FireEye, 2018).

In the above example,

**TCP** in the signature identifies which protocol it uses;

**\$HOME\_NET** refers to Source;

**\$EXTERNAL\_NET** indicates the destination;

**Any** indicates to the source port; and

**\$HTTP\_PORTS** refers to the default port for HTTP, which is 80.

Additionally, the direction (->) tells in which way the signature has to match. For instance, almost every signature would have an arrow to the right; this means that only packets with the same direction can match.

## 4.2 Anti-Virus & Malware

The attempt to test the response of the anti-virus and malware detection tools are described in this section. Herein, the protection theory is demonstrated into action that offers protection from viruses and malware. As previously outlined, the solution is effective and free of charge and the users do not have to be concerned about renewing the license or manually updating the anti-virus engine database signatures. For this purpose, Sophos Anti-Virus is used, which detects and manages various types of viruses, Trojans, worms, and malware. As such, it can be tasked for on-demand scans, on-access scanning, scheduled scans, logging and updating automatically (see Figure 44).

```
[sudo] password for hossamk:

Sophos Anti-Virus
=====
Copyright 1989-2019 Sophos Limited. All rights reserved.

Welcome to the Sophos Anti-Virus installer. Sophos Anti-Virus contains an on-
access scanner, an on-demand command-line scanner and the Sophos Anti-Virus
daemon.

On-access scanner           Scans files as they are accessed, and grants access
                             to only those that are threat-free.
On-demand scanner          Scans the computer, or parts of the computer,
                             immediately.
Sophos Anti-Virus daemon    Background process that provides control, logging,
                             and email alerting for Sophos Anti-Virus.
```

Figure 44. Anti-Virus scanning patterns.

In order to test the antivirus, the EICAR Anti-Virus test file can be used (Rubenking, 2013), which was first developed by the European Institute for Computer Antivirus Research (EICAR) and Computer Antivirus Research Organization (CARO). Although the file is a text file of between 68 and 128 bytes, it is a legitimate executable file, called a COM file. Thus, it can be executed via Linux kernel, and Windows (except for 64-bit version due to 16-bit limitations).

When executed, the EICAR test file will print "EICAR-STANDARD-ANTIVIRUS-TEST-FILE!" and then it will stop. The test string is written by noted anti-virus researchers Padgett Peterson and Paul Ducklin and engineered to consist of American Standard Code for Information Interchange (ASCII) human-readable characters, easily created using a standard computer keyboard. It also takes advantage of self-modifying code to work around technical issues that this constraint imposes on the execution of the text string (Sarang, 2018).

The main principle behind using the EICAR test file is that using real viruses for testing or demonstration purposes poses highly unacceptable risks in case the test fails. Thus, the EICAR test file can be used as it can be safely passed around and is non-viral; at the same time, the anti-virus software will react to it as if it were a virus. Thus, testing with EICAR can give meaningful results, without the unappealing and unacceptable risks (Stolfo, 2011).

The anti-malware test file is first downloaded to the local host machine to check the accuracy of the anti-virus and malware tool. The file format can be selected to be downloaded in different formats, as shown in Figure 45, such as .zip, .txt, and .exe.

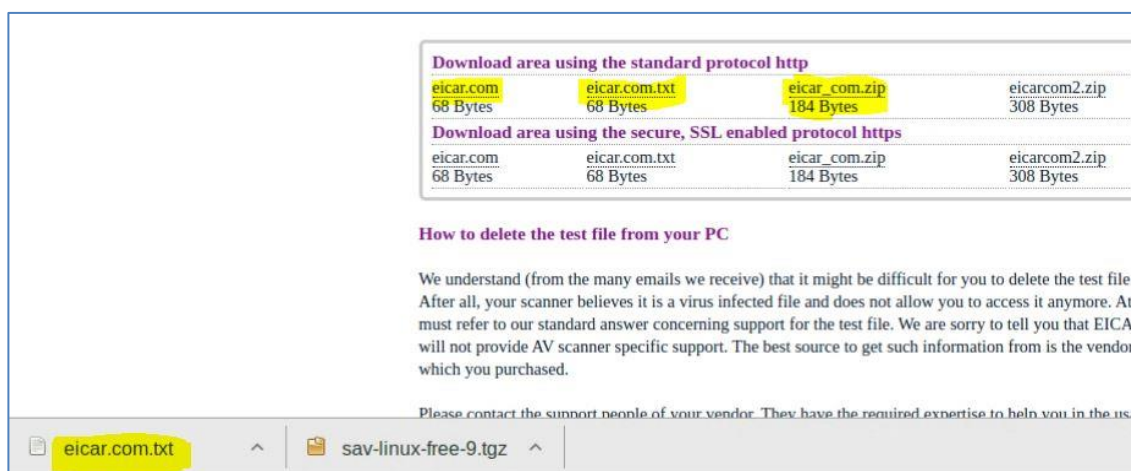


Figure 45. Anti-Virus test file in multiple extensions.

For this research, the file 'eicar.com.txt' was downloaded to the host. When the researcher attempted to open it, a message appeared (see Figure 46) with warning notification reading: 'Threat "EICAR-AV-Test" detected in the file. Access to file has been denied.'



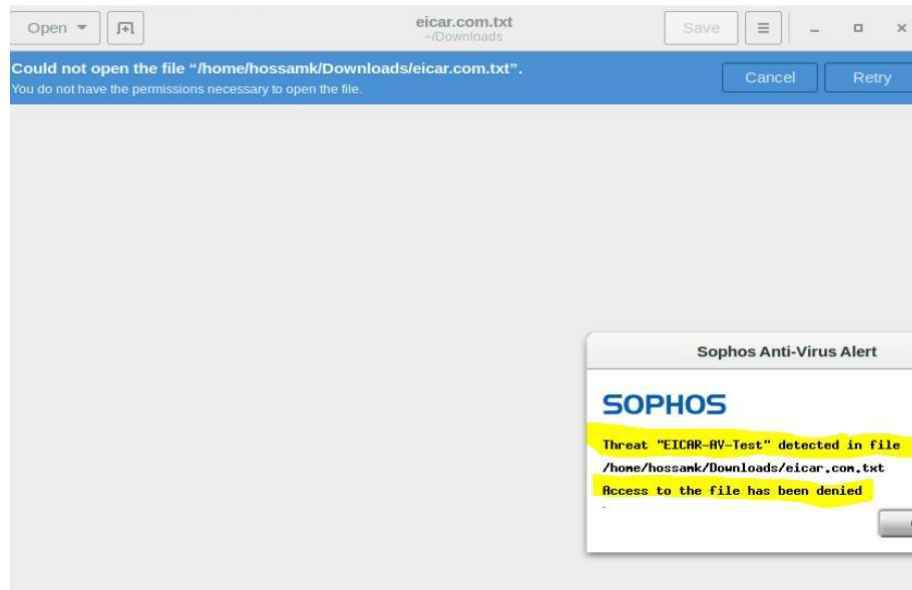


Figure 46. Access Denied for the EICAR-Test File.

Thus, the test proves to be successful in terms of accuracy and functionality as the anti-virus is able to detect threats types like viruses and malware.

#### 4.3 DNS Sinkhole & Advertisements Blocker

The Pi-Hole behaves like a Domain Name System (DNS) server, located between ISP (Internet Service Provider) and the Internet consumer. As such, its primary function is to intercept any outbound or inbound DNS requests, and it can block or allow specific domains. As can be seen in the figure below, the dashboard consists of total queries, queries blocked, percent blocked, and domains on a blacklist. For instance, the status shown in this case are:

- Total queries: logs belong to A (IPv4) and AAAA (IPv6) records - (64).
- Queries Blocked: the DNS blocked requests for IPv4 and IPv6 - (0).
- Percent Blocked: the percentage of total request DNS queries blocked from overall DNS requests - ( 0.0%).
- Domains on Blocklist: the website domains which are categorized as blacklisted - (113,448 URL domains).

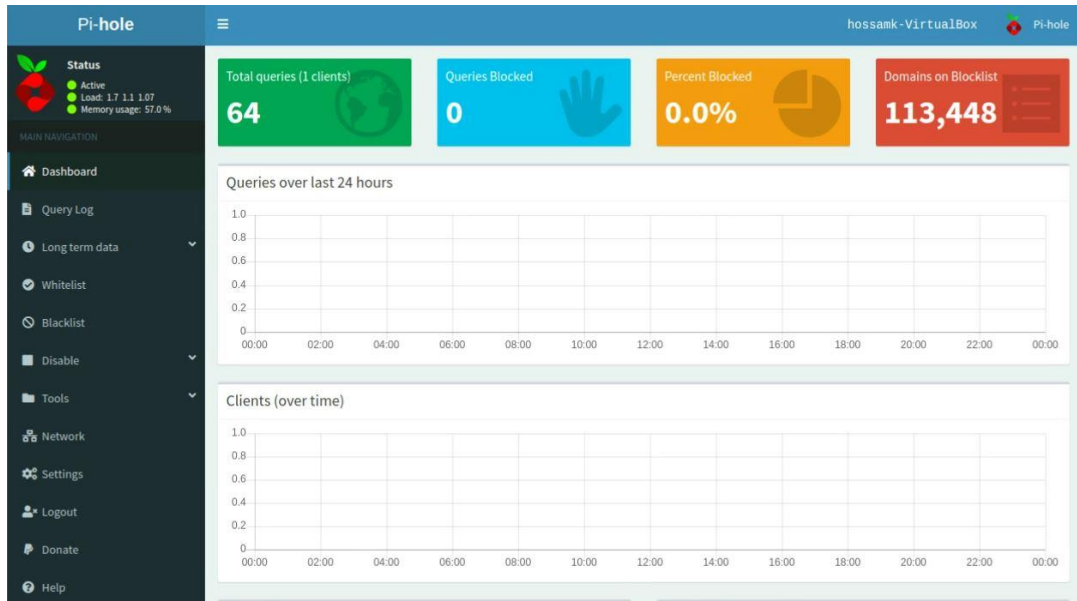


Figure 47. Pi-Hole Dashboard.

Pi-Hole, with third-party blocklists, blocks over 100,000 domains, as seen in Figure 47. These blocklists are trusted, and users would rarely have any false positives. However, most Pi-Hole users may want to expand the list (e.g., for known malware/ransomware hosts). For this purpose, the Pi-Hole community is an excellent source as several users have created and continue to maintain custom expanded blocklists, as shown in Figure 48. Hence, an expanded list is added for the sake of this research.

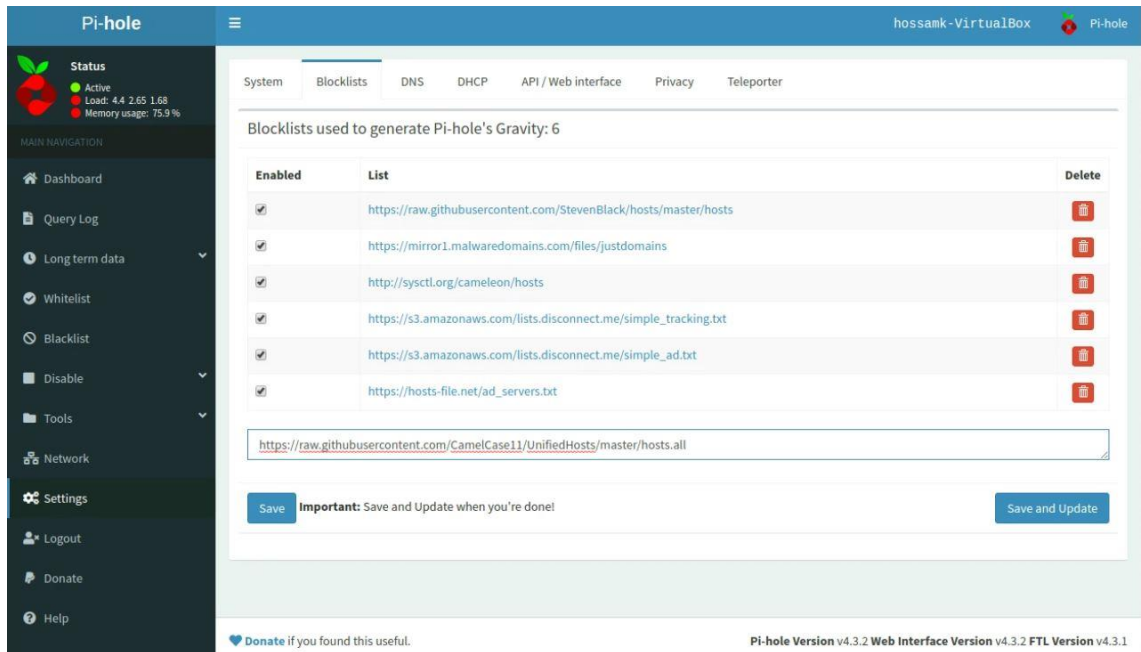


Figure 48. Pi-Hole Blacklist gravity.



After adding a hostname list that contains an additional blacklisted URL domain, the user can click 'Save and Update.' When this action was performed, the blacklisted domain count was seen to have increased from 113,448 to 1,475,563 URL domains, as shown below in Figure 49.

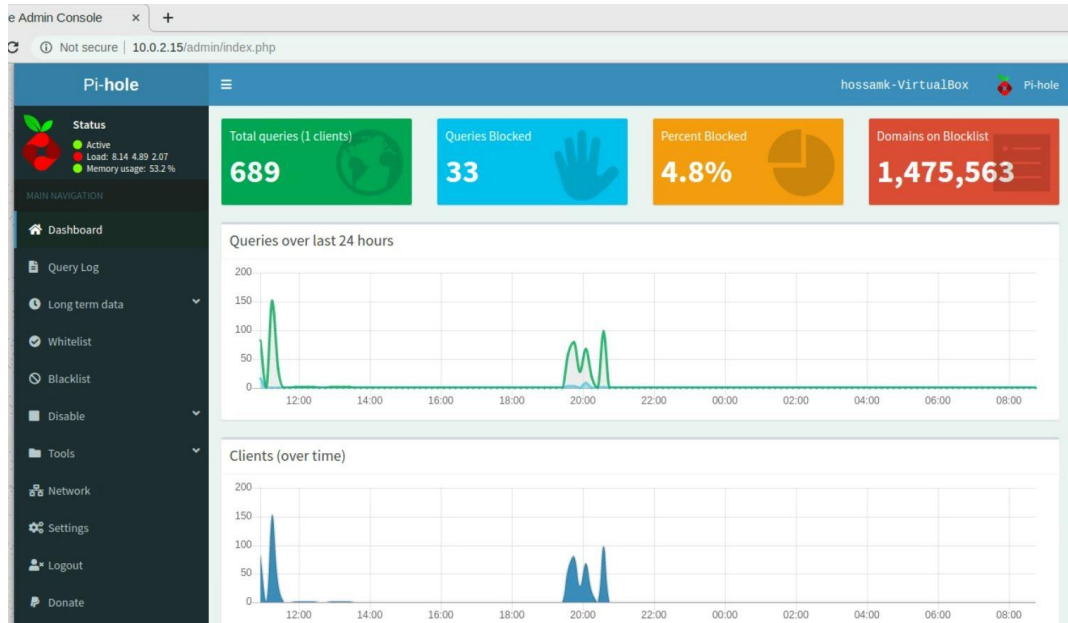


Figure 49. Pi-Hole Dashboard after updating Domains on Blacklist.

Additionally, for testing Pi-Hole ad blocker's functionality, users can visit a webpage that contains ads popping up when browsing; the functionality can be tested by checking if the ads are still displayed with the Pi-Hole tool being active. For such testing purposes, many websites can be used, such as Forbes.com, Businessinsider.com, BBC.com, etc. (see Figure 50). The test shows that the number of queries blocked on the dashboard of Pi-Hole increases by browsing the mentioned websites that contain ads. Therein, the Pi-Hole DNS inquires the domains and the ad-block feature compares the request to its domains on the blacklist.

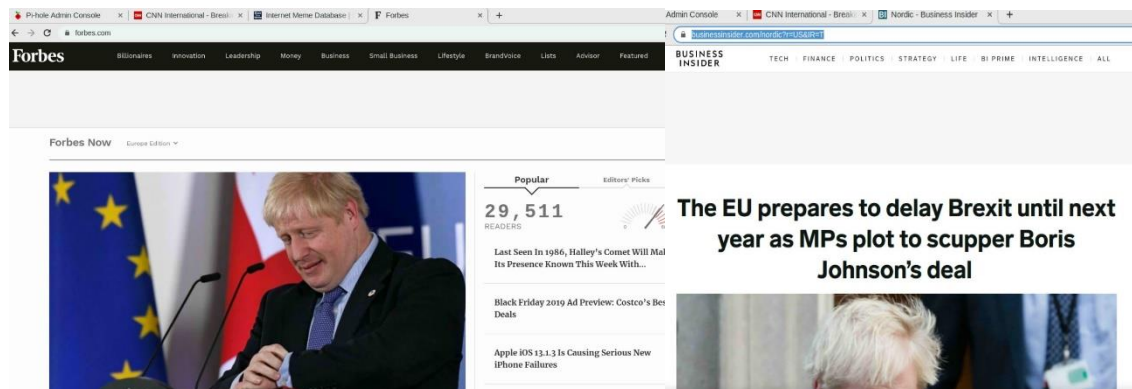


Figure 50. Pi-Hole testing websites for ad-block feature.

By navigating the Recent Queries to show the current logs for the Ad-Block feature and taking action against each browsed web domain, the test resulted in blocking ads of Forbes.com. For example, the blacklisted domains shown in Figure 51 below include:

- i. i.forbesimg.com
- ii. thumbor.forbes.com

Time	Type	Domain	Client	Status	Reply	Action
2019-10-22 09:08:21	A	encrypted-tbn2.gstatic.com	localhost	OK (forwarded)	IP (75.3ms)	Blacklist
2019-10-22 09:08:21	A	encrypted-tbn1.gstatic.com	localhost	OK (forwarded)	IP (79.3ms)	Blacklist
2019-10-22 09:08:21	A	encrypted-tbn3.gstatic.com	localhost	OK (forwarded)	IP (79.3ms)	Blacklist
2019-10-22 09:08:21	A	encrypted-tbn0.gstatic.com	localhost	OK (forwarded)	IP (81.5ms)	Blacklist
2019-10-22 09:08:20	A	thumbor.forbes.com	localhost	OK (forwarded)	CNAME (218.1ms)	Blacklist
2019-10-22 09:08:20	A	i.forbesimg.com	localhost	OK (forwarded)	CNAME (219.4ms)	Blacklist

Figure 51. Pi-Hole Recent Queries.

Thus, this test proved that the Pi-Hole functions as it should by blocking pop-up ads that can be regarded as a kind of threat as well. However, if the user visits legitimate sites such as Forbes, they can whitelist the ads, instead of blacklisting, as some websites may request revenue ads to be whitelisted to view the website content.

## 5 Conclusions & Future Considerations

The research and implementation of the proposed Internet home security solution was a valuable and exciting process, as it consisted of different layers of security that bond with each other. Moreover, the use of an open-source platform and software was reliable and stable, which helped accomplish the mission of testing all network security requirements, and validate whether it can achieve the purpose of securing the home Internet.

This thesis aimed to provide an affordable solution that can replace traditional commercial security solutions with the proposed Internet home security solution. As explained in the report, the proposed solution offers automated security protection level, with no commercial license being required, and is an embedded network monitoring system.

Based on the previous chapter, it is further concluded that the proposed solution is durable and functional from all aspects, i.e., each security measure was successfully checked against a particular threat and attack. Moreover, measuring performance versus functionality to check the output results proved satisfactory, i.e., the research requirement of securing the network with less human involvement and automated process, for behaving as an identity and alerting system, was successful. Furthermore, most of the security challenges listed in the introduction section of this paper were addressed by the proposed solutions, from both a theoretical and practical perspective. However, there were certain challenges that posed difficulty in overcoming. While some were eventually handled, some of them still remain that are yet to be addressed, which in other words may also be regarded as limitations of the research and/or the proposed solution. For instance:

- Lack of availability of a platform that gathers all security measurements in one system posed difficulty in practical implementations.
- The Internet home security solution requires some technical skills for implementation, which may limit its usage amongst home users.
- Users are likely to trust commercial solutions over open-source solutions; this would limit the reach of this solution.
- Lack of the security system to be able to fully auto-update itself is a limitation.

- Another limitation is that human interaction and intelligence is still required to avoid false positive alarms and malicious indicators.

Thus, to address these inherent limitations, and as part of the future study direction, the following key points are outlined in this conclusion:

- Future considerations can include encouraging home users to learn more security skills and test a variety of available open-source applications and tools; this would thereby improve their dependability on other commercial solutions for defending their home Internet environment from a malicious variety of attacks and vulnerabilities.
- The most critical feature for keeping and preventing attackers from compromising security measures is by automating the process of updating the security software, database, and attack signatures, which further studies can explore.

Finally, as part of recommendations for home users, experimenting and educating oneself with future contributions in the field of Internet home security can introduce users to the newly available security systems. As these options tend to be readily available on the Internet, users can fully secure their home Internet from threats. For instance, fully integrated built-in Network Security Monitoring (NSM) solutions such as Security Onion are currently available in the market (Security Onion Solutions, 2019). This particular product is a Linux distribution system encapsulated with various security sensors and tools, including Unified Threat Management (UTM) solution, log management, enterprise security monitoring, full packet capturing and network protocol analyzer. Nonetheless, since it is a comprehensive tool combining all these features in one system, a massive overhead of knowledge might be needed to be able to manage it.

## References

Ashford, W. (2015). Rapid7 research exposes Internet of things security problems. [online] ComputerWeekly. Available at: <https://www.computerweekly.com/> [Accessed 29 Oct. 2019]

Bejtlich, R. (2008). Network security monitoring: Know your network. [online] SearchITChannel. Available at: <https://searchitchannel.techtarget.com> [Accessed 20 Mar. 2019].

Catchpoint. (2015). Ad-blocking on Apple iOS9: valuing the end user experience. [online] Available at: <http://blog.catchpoint.com/2015/09/14/ad-blocking-apple/> [Accessed 15 Jul. 2019].

Check Point. (2018). Live cyber attack threat map | Check Point Software. [online] Available at: <https://threatmap.checkpoint.com/ThreatPortal/livemap.html> [Accessed 5 Nov. 2018].

Connolly, K. (2016). Packet analyzer. [online] En.wikipedia.org. Available at: [https://en.wikipedia.org/wiki/Packet\\_analyzer](https://en.wikipedia.org/wiki/Packet_analyzer) [Accessed 21 May. 2019].

Crocker-White, C.. (2019). Deploy network-wide ad-blocking with Pi-hole and a Raspberry Pi. [online] Available at: <https://www.balena.io/> [Accessed 15 Jul. 2019].

DNSlytics.com. (2018). AS/BGP report for the country Finland.. [online] Available at: <https://dnslytics.com/bgp/fi> [Accessed 7 Nov. 2018].

Diamond Systems Corporation. (2017). Single-board computer. [online] [whitepaper.opsy.st](http://whitepaper.opsy.st). Available at: <http://whitepaper.opsy.st/> [Accessed 21 May. 2019]

FireEye. (2018). Cyber threat map. [online] Available at: <https://www.fireeye.com/cyber-map/threat-map.html> [Accessed 5 Nov. 2018].

Forcepoint. (2019). What is an Intrusion Prevention System (IPS)?. [online] Available at: <https://www.forcepoint.com/> [Accessed 25 Mar. 2019].

- Juniper Networks. (2018). Understanding port mirroring - TechLibrary - Juniper Networks. [online] Juniper.net. Available at:<https://www.juniper.net/> [Accessed 26 May 2019].
- Mojidra, N. (2016). Stateful vs Stateless Firewalls - Cybrary. [online] Cybrary. Available at: <https://www.cybrary.it/0p3n/stateful-vs-stateless-firewalls/> [Accessed 21 May 2019].
- Oracle Corporation. (2019). Oracle VirtualBox - User Manual. [online] Available at: [https://www.virtualbox.org/manual/ch06.html#network\\_internal](https://www.virtualbox.org/manual/ch06.html#network_internal) [Accessed 12 Jun. 2019].
- Robbie, C. (2013). BriarIDS - A home intrusion detection system (IDS) solution for the Raspberry Pi. [online] Available at: <https://briarids.wordpress.com/about/> [Accessed 6 Apr. 2019].
- Rubenking, N.J. (2013) Is your antivirus working? [online] PCMag. Available at: <https://pcmag.com/opinion/73568/is-your-antivirus-working> [Accessed 2 Nov. 2019]
- Rubens, P. (2015). 10 Open Source Security Breach Prevention and Detection Tools. [online] Esecurityplanet.com. Available at: <https://www.esecurityplanet.com/> [Accessed 18 Sep. 2019].
- Sanders, C., Smith, J. and Bianco, D. (2014). Applied network security monitoring. Waltham, MA: Syngress, an imprint of Elsevier.
- Sarang, R. (2018). Trending: IoT malware attacks of 2018. [online] Available at: <https://securingtomorrow.mcafee.com/> [Accessed 5 Nov. 2019].
- Security Onion Solutions. (2019). Security Onion Documentation. [online] Available at: <https://securityonion.readthedocs.io/en/latest/about.html> [Accessed: 25 Mar. 2019].
- Stanislav, M. and Beardsley, T. (2015). Hacking Sousi: a case study on baby monitor exposures and vulnerabilities. [online] Rapid7.com. Available at: <https://www.rapid7.com/> [Accessed 8 Jan. 2019].
- Stolfo, S. (2011). Insider attack and cybersecurity. New York: Springer, p.1.
- Veríssimo, P. and Rodrigues, L. (2009). Distributed systems for system architects. Boston: Kluwer Acad. Publ., p.377.