Feasibility Study on using Single Board computers for Intrusion detection and penetration testing in IoT Environment

---

Dhiyazen Ali Al-Awaisi

K00191943

A Final Year Project submitted in partial fulfilment of

The requirements of Limerick Institute of Technology

For the degree of Bachelor of Science (Honours) in

Computer Networks and Systems Management

Supervised by:

Mike Winterburn

May,2019

Ethical declaration

I declare that this project and document is wholly my own work except where I have made explicit reference to the work of others. I have read the Department of Information Technology Final Year Project guidelines and relevant institutional regulations, and hereby declare that this document is in line with these requirements.

I have discussed, agreed, and complied with whatever confidentiality or anonymity terms of reference were deemed appropriate by those participating in the research and dealt appropriately with any other ethical matters arising, in line with the LIT Research Ethics Guidelines for Undergraduate and Taught Postgraduate Programmes policy document.

_____ _____

[Type name here] [Date]

Acknowledgment

I would like to express my appreciation to everyone who provided me with the help to complete this Final Year Project. A special gratitude I give to our final year project supervisor, Mr. Mike Winterburn, whose help in giving suggestions and encouragement, helped me to coordinate my project especially in completing my final year project.

Table of Contents

Table of Figures

Table of Tables

Abstract

This is a feasibility study and the implementation on the possibility of using single board computing devices like Raspberry pi and Banana Pi M64as an intrusion detection system and penetration testing system to increase the network environment system. The focus of this study was to investigate if these devices can handle intrusion detection software and remote penetration testing and then compare the two devices performance. To examine this a virtualized test environment was built containing two virtual machines mimicking a raspberry pi and a banana pi and another three virtual machines mimicking a small network environment. Tests measuring the network throughput, CPU usage and RAM usage were performed on the Raspberry Pi VM and Banana Pi M64 VM, each of them running the operating system Kali Linux. The results of these tests were that both devices could be used as an intrusion detection system and a remote penetration testing system but has some limitations that could impede usage depending on the requirements of the user and environment. The Banana Pi M64 show benefits of its updated hardware by suffering lower throughput degradation than Raspberry Pi, while using less of its total CPU and RAM capacity.

List of abbreviations

CPU	Central Processing Unit
CRC	Cyclic Redundancy Checks
DDoS	Distributed Denial of Service
GUI	Graphical User Interface
HDD	Hard Disk Drive
ICMP	Internet Control Message Protocol
IDS	Intrusion detection system
IoT	Internet of things
IP	Internet Protocol
IPv4	Internet Protocol version 4
OS	Operating system
Pen-testing	Penetration testing
RAM	Random Access Memory
RFID	Radio Frequency Identification
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VDS	VMware vSphere Distributed Switch
VM	Virtual Machine
WSN	Wireless Sensor Networks

Introduction

This project is a feasibility study and the implementation of a virtualized environment on the possibility of using single board computing devices like Raspberry pi and Banana Pi M64 as an intrusion detection system and penetration testing system to increase the network security in an IoT environment. The focus of this study was to investigate if these devices can handle intrusion detection software and remote penetration testing and then compare the two devices performance. To examine this a virtualized test environment was built containing two virtual machines mimicking a raspberry pi and a banana pi and another three virtual machines mimicking a small network environment. All the virtualization of the physical network was done in a private internal network within VMware ESXI virtualized environment. The Windows Server 2016 VM is fully configured with a firewall to imitate a real windows server instance, same system and firewall configuration was made to the Windows 10 VM and the Ubuntu VM. The Raspberry Pi VM was configured with 1GB of RAM and 32GB of storage space and it is using Kali Linux as an operating system. The Banana Pi M64 VM was configured with 4GB of RAM 64GB of storage space and it is using Kali Linux as an operating system.

Tests measuring the network throughput, CPU usage and RAM usage were performed on the Raspberry Pi VM, Banana Pi M64 VM, Windows 10 VM, Windows server 2016 VM and Ubuntu VM. The aim of the first test is to create a baseline of the performance of the VMs without any IDS or penetration testing software running on them. Network throughput, CPU usage and RAM usage are the most important metrics that need to be monitored during testing, because any overload can cause a very noticeable decrease in network efficiency and performance. For the throughput to be within acceptable limits a requirement was made that there should be no loss greater than 30% - 40% and the CPU Usage should not go above 55%. This is intended also as a point of reference used to evaluate and analyse the data gathered from the experiment.

There is a limitation to this project, the project will only focus on the prospect of using a Raspberry Pi and Banana Pi M64 as an intrusion detection system using the software Snort and Bro-IDS and test their ability in performing a penetration testing using different tools. This project will not look at specific rules, options for Snort. On the other hand, when it comes to penetration testing tools this project will only cover the scanning and uncovering of possible exploits but will not cover the use of any exploits.

The results of these tests were that both devices could be used as an intrusion detection system and a remote penetration testing system but has some limitations that could impede usage depending on the requirements of the user and environment. The Banana Pi M64 show benefits of its updated hardware by suffering lower throughput degradation than Raspberry Pi, while using less of its total CPU and RAM capacity. it was also concluded that the price is one of the strongest arguments for this kind of implementation. The low cost of the Raspberry Pi or the Banana Pi means that it can be a more affordable IDS compared to what would otherwise be an expensive enterprise grade solution. This solution opens several possibilities for users in increasing their network security from intrusions using low priced and affordable single board computing devices. At the end, it is advisable to use the Banana Pi M64 for pen-testing and as an IDS as it can handle more processing than the Raspberry Pi.

Literature Review Chapter

1 Introduction

Internet of Things (IoT) is term was first invented and used in 1998, it refers to uniquely identifiable objects, things and their virtual representations in an internet-like structure (Ashton, 1998). Nowadays, IoT is a well-known terminology and a rising trend in the development of IT arena, there has been no agreed terminology by the community and the developers until now. Now, there is a single unified definition created by the international telecommunication union for the term IoT "Global infrastructure for the society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies." (ITU, 2012). At the same time, there is many alternative definitions for the term IoT. Many of these definitions focus on the technologies which became connected to the IoT. The rest of the definitions focus on internet-related aspects of the IoT, such as network technologies and internet protocols. And a third type centres on the semantic challenges in the IoT relating to storage, search, data mining and organization of large volumes of information (atzori, et al., 2010).

Nowadays, the increase growth of IoT environments has been increasing rapidly along with various applications and technologies e.g. the internet. As far as the growth of Internet of Things, Gartner, a technology research and advisory corporation expects that there will be about 20.8 billion IoT devices by the 2020 which is expected to make an evolution in government, transportation, education, and finance (Gartner, 2016). However, the existence of such a large networks and environments of IoT will pose new security and trust threats that put all those IoT environments at a high risk, thus harming the affiliated users and organizations using this technology.

Internet is the core component if IoT because almost all the security threats that threatens the Internet also threatens the IoT as well. Moreover, the fast development, use and wider adoption of IoT devices in our lives signifies the importance of addressing these security threats before the full deployment of the IoT technology (Chrysostomou, et al., 2015). Although some of the companies involved in the IoT arena state that their technologies are secured and protected but they are still prone to various types of attacks. Since all of the devices in the IoT are interconnected and they have a direct impact on the lives of users, there is a need for a well-defined security threat classification and a proper security

infrastructure with new systems and protocols that can mitigate the security challenges regarding privacy, data integrity, and availability in IoT (Hadjichristofi, et al., 2015)

2 IoT Overview

IoT has wan the massive attraction of developers, inventors and global organizations recently because of the expansion of appliances connected to the Internet (atzori, et al., 2010) (Whitmore, et al., 2015). IoT simply means the interconnection of vast and diverse network frameworks and systems in different patterns of communication (Sardana & Horrow, 2012) (Ala Al-Fuqaha, et al., 2015). Moreover, the IoT is a massive domain where physical items are constantly integrated to form a network with the goal of providing advanced and smart services to users (Botta, et al., 2016) (Shancang Li, et al., 2014). Nowadays, IoT attached itself into almost every single aspect of networking, technology and electronics development and use, the figure shows the percentage of devices connected to IoT networks in the years 2014 and 2016 according to a study done in 2016 (SpiceWorks, 2016):

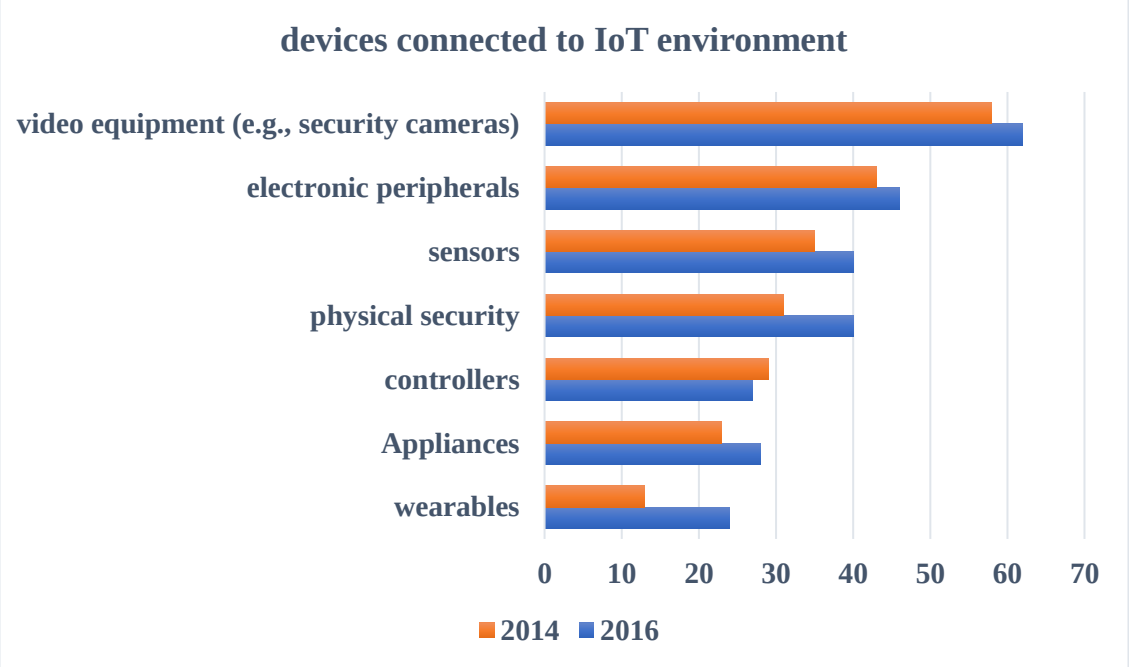


Figure 1 Number of devices connected to IoT environment

3 Main IoT technologies

IoT is implemented using a variety of existing network technologies, and more specifically using the following three:

I. Radio Frequency Identification (RFID):

RFID technology uses radio frequencies which enables the microchips to transmit data in a wireless communication. Because RFID allows things to be identified and enables communications between them, it enables applications to be everywhere. As a result, RFID is one of the key technologies that the IoT depends on (CoreRFID, 2017).

II. Wireless Sensor Networks (WSN):

WSN is a network formed by substantial number of geographically distributed autonomous, low cost and low power sensors nodes where each node is equipped with a sensor to detect any phenomena at the physical layer. (Commission, n.d.)

III. Cloud Computing (Cloud IoT):

It is a mesh of IoT physical implementation and Cloud Computing service that allows you to easily and securely connect, manage, and analyse data from substantial number of devices connected to the Cloud-IoT. Cloud IoT provides a solution for collecting, processing, analysing, and visualizing IoT data in real time which increases the performance and stability of the devices connected to the Cloud IoT.

4 IoT Structure

IoT is typically structured into three main layers; the physical layer, network layer, application layer (Song, 2013).

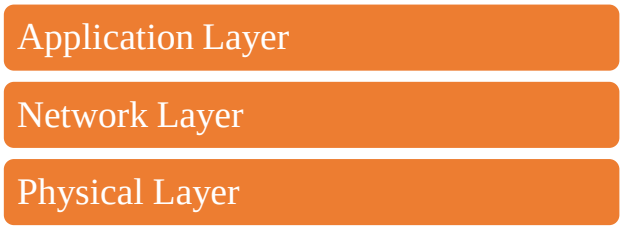


Figure 2 IoT Layers

I. Physical Layer:

The physical layer basically responsible for the physical interconnected devices and its main purpose is to perform device identification and service discovery to the devices connected to the IoT Network. These devices can many of various types e.g. raspberry pi, but to be considered as IoT devices they need to utilize radio or wireless communication technology that allow them to interconnect with each other (Fremantle, 2010).

II. Network layer:

The network layer has the function of managing wireless and wired connections. This layer includes network interfaces, communication channels, network management and intelligent network flow processing (Yang, et al., 2012). It transfers the collected data through the sensors, and computers across the interconnected wired and wireless networks (Hinai & Singh, 2017).

III. Application layer:

The application layer is the interface between the applications in the devices and the end users. It is responsible for the communication between them. It can support various services required by business (Hinai & Singh, 2017). It can store data into a database providing storage capabilities to the collected data (Khan, et al., 2012).

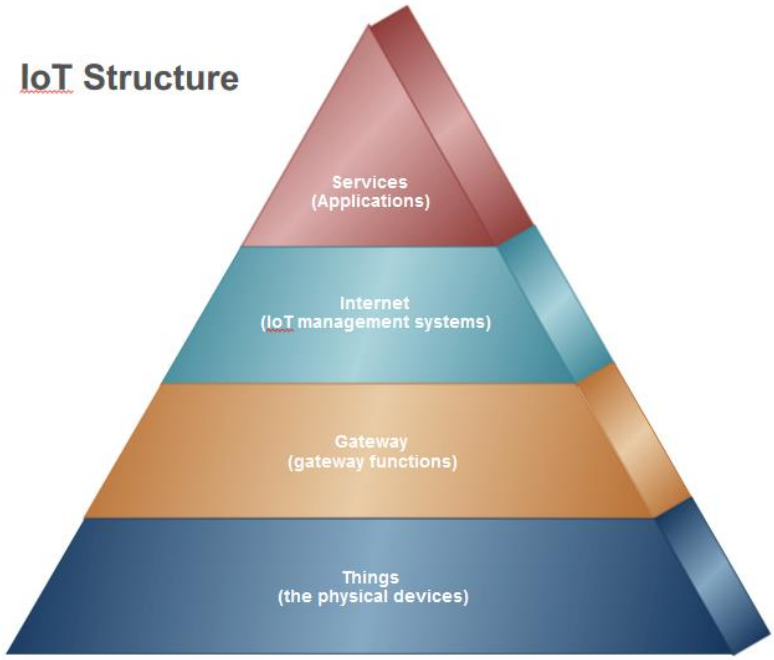


Figure 3 IoT environment structure

5 IoT Security

IoT is considered a new emerging technology in the field of IT, so there is a high and an increasing demand from the developers and users to define its security requirements. The difficulty in defining the security requirements is that IoT is an implementation of network technologies and an integration of existing network infrastructures thus, all the challenges of the classic network technology are passed by default onto the IoT environments that have these technologies embedded into them. In addition, there is an additional security threats that arise from the conjunction and combination of the various technologies and the open standards and protocols established for the IoT. The most desirable security aim of IoT is to protect the data collected, because the data collected from the physical devices may also contain sensitive user information.

5.1 Security requirements

Security comprise of all the approaches and procedures that aim to restore, preserve and guarantee the protection of information in a network environment or a computer system from malicious attacks. News publisher's puts security at the top of concerns: leakage of personal data and economic espionage, infection of sensitive computer systems, identity theft and fears about card payments.

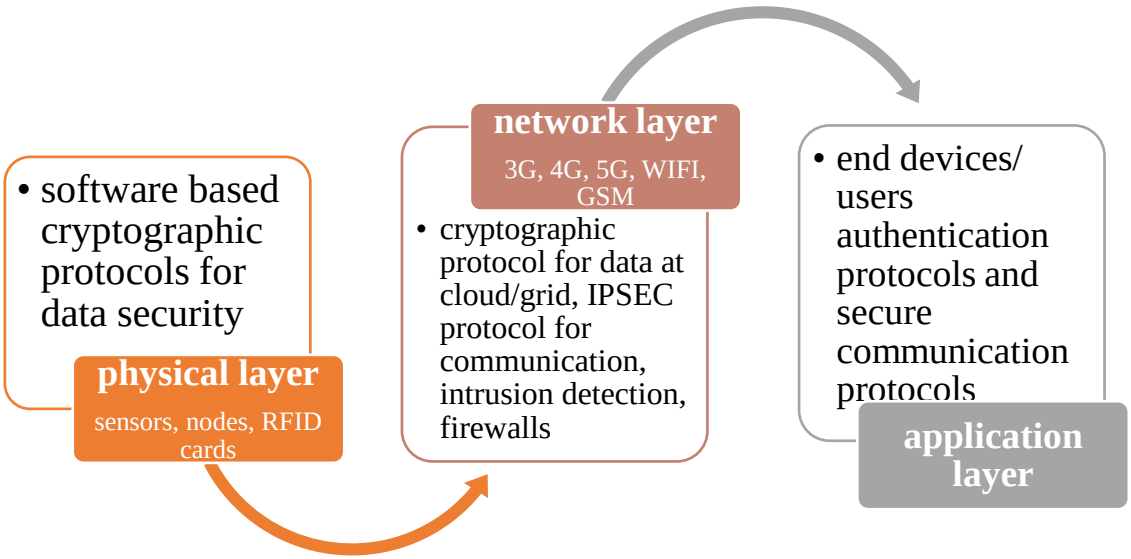


Figure 4 classic IoT layers with classic security requirements

The security of computer networks and information systems in general, consists to provide the following services (noura, 2016):

I. **Confidentiality:** It ensures that information is made unintelligible to unauthorized individuals, entities, and processes. The confidentiality is applied in IoT devices and

environments by making sure that the exchanged data and queries between the devices, the nodes and control systems are not disclosed by third unauthorized entities.

II. **Data Integrity:** It ensures that data has not been modified by a third party (accidentally or intentionally). Error detection techniques must be provided at each device and node, to ensure no tampering of data occurs. Other techniques and procedure like Cyclic Redundancy Checks (CRC), Checksum, and Parity Bit are preferred for their low resource's requirement, but for more secure and aggressive error detection methods such as WH cryptographic hash function should be applied (Kaps, 2006).

III. **Authentication:** It verifies that the data source is the pretended identity.

IV. **Non-repudiation:** It ensures that the sender of the message cannot deny having sent the message in the future.

V. **Availability:** It ensures that the services of the system should be available for legitimate users.

VI. **Privacy:** It ensures that users' identities should not be identifiable nor traceable from their behaviours and their performed actions in the system.

5.2 Security and Privacy issues and concerns

For several years, IoT developers and users have raised privacy and security concerns related to IoT devices (Wolff, et al., 2017). The main privacy concern of IoT devices is that they collect sensitive information, including financial account numbers, health information and more.19 Experts have identified two main types of sensitive personal data (Koo, December 27, 2017). The prime security concern of the IoT is the vulnerability, of devices to external and internal cyber-attacks, such as the May 2017 WannaCry attack21, the October 2016 Distributed Denial of Service (DDoS) attacks22 that targeted the IoT environments only (Kirtley & Memmel, 2018). The attackers were exploiting the vulnerability of one device, making it the "master," which then identifies and targets other vulnerable devices, networks, and systems (Kirtley & Memmel, 2018).

In 2018, new areas of security and privacy concerns have arisen, including smart transport systems, smart manufactures and smart cities, as well as medical devices. Additionally, some users and developers noted that IoT devices were beginning to be used more frequently in a new form of domestic abuse in which abuser's target and harass their

partners or former partners by changing the settings of a smart thermostat, playing bursts of loud music through virtual assistant devices, and changing the lighting throughout their home, among other actions.

5.3 IoT Security challenges

Because IoT is a new and still developing technology, there is a need to find and solve many of the security challenges IoT is facing. In IoT there are many security challenges that need to be addressed Table 2 shows some security challenges IoT facing in different sectors.

	Applications				
challenges	Smart grids	EHealth	Transportation systems	Smart cities	Manufacturing
Resources constraints	+	+++	-	+++	+
Scalability	+++	++	+++	+++	++
QoS constraints	++	++	+++	+++	++
Data management	++	++	+	+++	+
Lack of standardization	+++	+++	+++	++++	+++
Amount of attacks	+++	+++	+++	++++	+++

Table 1 IoT security challenges

But the two main issues IoT is facing are Lack of standardization and Diversity of attacks;

I. Lack of standardization:

There is no existing standard protocol that is adopted by all the Supervisory control and data acquisition (SCADA) based IoT systems and this lack of standardization is negatively affecting the growth of IoT, and IoT security. There are about 150 to 200 open standards (Miller & Rowe, 2012) but majority of them are created by the users and developers rather than a trusted standards issuer.

II. Diversity of attacks:

IoT networks and exposed internally and externally to all kinds of attacks which harm the integrity and the security of the IoT network. Since all the data in IoT is conveyed through wireless media IoT networks are more vulnerable than regular systems.

5.4 IoT Security Strategies

The security aspect in IoT environments has been tackled and viewed from different points of view. Various proposals have been proposed to emphasize the importance of security in the various IoT layers. M. Asplund and S. Nadjm-Tehrani (2016) both described the significant risks related IoT security and the rise of new different devices and platforms involved in the system and the inefficiency of the old defense strategies to tackle.

The distinctive validation of the mass range of IoT devices, the need of secure, energy efficient and low cost authentication schemes still emerging (Ray, et al., 2016).A new secure multi-hop transmission relay to avoid the eavesdroppers was proposed in the journal article of Security Enhancement for IoT Communications Exposed to Eavesdroppers With Uncertain Locations (Xu, et al., 2016).

Guo et al. tried to increase security by adding biometrics into the IoT system to prevent any unauthorized access to the network or the system. Similar method was tried out by Hossain et al. (2016) in multiple levels for improving the secure transmission using Reed Solomon Codes for error detection (Giuliano, et al., 2017), secure key renewal approach (Chen, et al., 2017) security approach for context aware IoT systems (Sedjelmaci, et al., 2017) was also experimented with. A lightweight secure anomaly detection algorithm using Nash Equilibrium concept especially suited for resource constrained IoT devices was proposed by Sedjelmaci.

A less power consuming and negligible overhead algorithm which is a modification of the AES algorithm which prevents leakage of stored key using false key-based advanced encryption standard (AES) (Yu & Köse, 2017), (Thapliyal, et al., 2017), (Park, et al., 2017). Also, cryptographic algorithm for preventing attackers from stealing the secret key was also proposed (Zhou, et al., 2017).

A physical level security using FPGA for increasing the longevity of the smart devices was experimented which was claimed To be more flexible in hardware level implementation compared to other cryptographic algorithms (Dao, et al., 2017), (Zhang,

et al., 2017) .It is evident in the literature that the use of ECC gives secure solutions which are cost effective and with less overheads with respect to IoT devices (Qiu, et al., 2017).

5.5 Categorization of IoT layers attack

IoT is implemented using various preexisting network technologies. Table 1 shows the main attacks on the IoT layers.

Physical layer attacks	Network layer attacks	Application layer attacks
Node tempering	Traffic Analysis Attacks	Virus and Worms
RF interface	RFID Spoofing	
Node jamming	RFID Cloning	Spyware and Adware
Malicious Node Injection	RFID Unauthorized access	
Physical Damage	Sinkhole Attack	Trojan horse
Social Engineering	Man-In-The-Middle-Attack	
Sleep Deprivation Attack	Denial of Service	Malicious scripts
Malicious Code Injection on the Node	Sybil Attack	Denial of Service

Table 2 examples of attacks on IoT layers

Since IoT uses less secure technologies, there is a need for a proper categorization of the attacks, so that better counter measurements can be created, developed and implemented for increasing the level of security in IoT environments (Perrig, et al., 2004). IoT attacks are categorized as follows:

I. Physical attacks

Physical attacks are focused on the hardware components of the IoT system and the attacker needs to be physically close to IoT system or the IoT environment. The physical components of IoT systems, such as controllers, RFID readers etc… are vulnerable to different physical attacks (Symantec, 2016).

II. Network attacks

These attacks are more focused on the IoT system network and the attacker does not necessarily need to be close to the network for the attack to work (Hezam, et al., 2018). IoT is vulnerable to enormous amount of attacks because unlike traditional internet stack design, IoT system has its own network stack. The IoT stack demands lightweight

protocols such as 6loWPAN and IEEE 802.15.4 different from the standard conventional internet protocols and less secure.

III. Application attacks

Application attacks are the main source of security vulnerabilities in any IoT environment. Application attacks exploits the system by using Trojan horse programs, worms etc... Can steal information, tamper with data, deny service and even harm the devices in the IoT System.

6 Penetration testing and vulnerability assessment

Due to the many challenges IoT is facing, developers started to employ offensive attack techniques to discover vulnerabilities within the IoT system. Because malicious attacks need only a single vulnerability exploit to be successful. Security researchers use automated tools to carry out these simulated attack techniques (Chen, et al., 2018).

6.1 Penetration testing vs vulnerability assessment

A vulnerability assessment uses commercial and public open tools to scan hosts and devices for common weaknesses, vulnerabilities and misconfigurations at the network layer or the application layer. This is usually an acceptable method for scanning many hosts and devices to gather and analyze data to create a baseline overview of the general security level, but the test is limited to only conducting checks for which the creators of the tools used have released plugins. If the scanning tool doesn't know about it, then it won't be picked up (Shinde & Ardhapurkar, 2016).

On the other hand, Penetration testing is a method used to ensure that vulnerabilities or weaknesses in networked environments, web applications and physical layers are known about and can be addressed before they are exploited by an external threat. Penetration testing can contain many different areas, mostly covering internal networks and Internet-facing hosts.

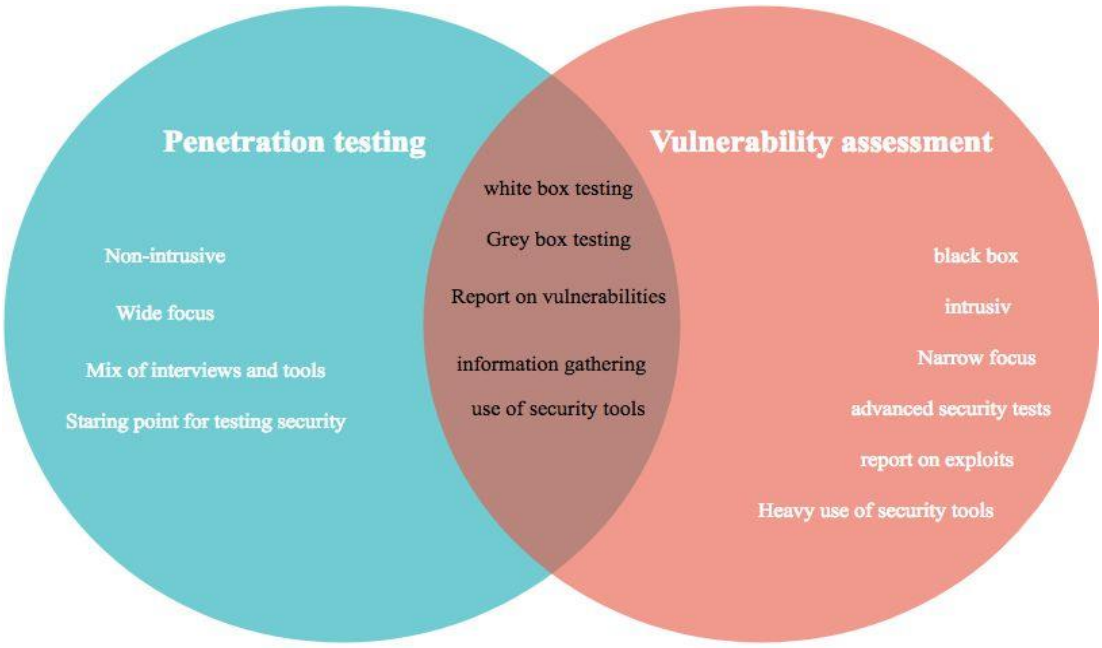


Figure 5 pen-testing vs vulnerability assessment

6.2 The reason of use

Vulnerability testing is performed because of the following reasons (Chunlei, et al., 2014) (Altaf, et al., 2015) :

I. To know the attack vectors of the attacker and how they can gain access to the system.
II. To know the highest risk and threat to the applications, so that they can be patched as soon as it is detected by the system.
III. Penetration testing is performed, as many automated software's are unable to detect the risk.
IV. Identifying the business risk that can be done through the exploitation of the product. For instance, Microsoft will lose faith of customers if their website is hacked and attacker gives free downloads on the internet.

6.3 Risk rating system

One of the most important pieces of information to help with penetration testing and vulnerability assessment is the risk rating. Generally, this is categorized as critical or high, medium, low or informational, but will also outline either the impact of the exploit on the host or network and show the likelihood of an attack or the ease of exploitation it poses

(Tang, 2014). Usually any critical or high-risk issue that is easy to exploit should be fixed first, as should any issues highlighted in the report that gave the testers any access to a host, data or information to which the company would not want an attacker to have access (Tang, 2014).

	consequences				
likelihood	Insignificant	Minor	Moderate	Major	Critical
Rare	**LOW** Accept the risk routine management	**LOW** Accept the risk routine management	**LOW** Accept the risk routine management	**MEDIUM** Specific responsibility And treatment	**HIGH** Quarterly senior management review
Unlikely	**LOW** Accept the risk routine management	**LOW** Accept the risk routine management	**MEDIUM** Specific responsibility And treatment	**MEDIUM** Specific responsibility And treatment	**HIGH** Quarterly senior management review
Possible	**LOW** Accept the risk routine management	**MEDIUM** Specific responsibility And treatment	**MEDIUM** Specific responsibility And treatment	**HIGH** Quarterly senior management review	**HIGH** Quarterly senior management review
Likely	**MEDIUM** Specific responsibility And treatment	**MEDIUM** Specific responsibility And treatment	**HIGH** Quarterly senior management review	**HIGH** Quarterly senior management review	**EXTREME** Monthly senior management review
Almost certain	**MEDIUM** Specific responsibility And treatment	**MEDIUM** Specific responsibility And treatment	**HIGH** Quarterly senior management review	**EXTREME** Monthly senior management review	**EXTREME** Monthly senior management review

Table 3 risk matrix (Office, 2018)

7 Intrusion detection overview

Intrusion detection is the process of identifying actions, patterns and unusual use of a system. The process, known as intrusion, aim to collect, analyze and obtain unauthorized access to a network or a computer system. The Intruder may be external or internal. Internal intruders are users inside the network and External intruders are users outside the targeted network trying to obtain unauthorized access to system information (Vacca, 2013) (Patel, et al., 2010).

7.1 Intrusion detection system (IDS)

A classic intrusion detection system (IDS) consist of sensors, network flow analysis engine, and reporting and logging system. Sensors are installed at different points in the network. The aim of the sensors is to gather host data and network flow data such as traffic statistics, packet headers, service requests, etc. The sensors then send the gathered data to the analysis software, which is responsible for investigating the collected data and detect unnormal patterns in the network. When the analysis software detects an intrusion, the reporting system generates an alert and notify the network administrators. IDS can be classified as either Network-based IDS (NIDS) or Host-based IDS (HIDS).

NIDS connects to one or more network segments and monitors network traffic for unnormal activities and unusual network flow patterns. HIDS is contained in a computer device and monitors traffic and the use of the system or the host for unnormal activities and unusual patterns of use. Unlike NIDS, the HIDS analyzes the network traffic and system calls, running processes, file-system changes, interposes communication, and application logs.

7.2 Previous research on IDS in IoT

Over the recent years, several review articles have been published on IDSs for technologies related to IoT such as Mobile Ad hoc Networks(MANETs) (Mishra, et al., 2004) (Anantvalee & Wu, 2007),wireless sensor networks (WSNs) (Hussain, et al., 2004) (Abduvaliyev, et al., 2013)and cyber-physical systems (Mitchell & Chen, 2014).

Mishra et al. (2004)point out that applying the research of wired networks to wireless networks is not an easy task due to the fundamental architectural differences, especially the lack of fixed infrastructure. The authors argue that the type of intrusion response for wireless ad hoc networks depends on the type of intrusion, the network protocols and applications in use, and the confidence in the evidence. The authors also present a detailed

discussion of seven IDSs proposals for MANETs according to the following methodologies: distributed anomaly detection and mobile-agent-based detection. In both cases, an IDS agent runs at each mobile node and performs local data collection and local detection. The difference between the two methodologies lies in the global detection: the distributed anomaly detection uses information from neighboring nodes to build a cooperative detection engine while the mobile-agent-based detection employs mobile agent's technology for intrusion detection and response.

Design and prototype chapter

8 Introduction

This chapter proposes a Virtualized environment built to study the possibility of using Intrusion detection systems and Penetration Testing software in Portable/Micro computing devices and then study the performance of these machines and compare them to the security appliances being used now days. Also, it is going to help in investigating the idea of the possibility of creating neural network with these devices and implementing them in a pre-existing network infrastructure

The system design will be built in VMware ESXi virtualized environment. Networking aspects can be simulated and built-in by using User Defined group ports and distributed switches which will make the flow of data like the real physical scenario, and it will give the ability to real-time packet and data flow tracking which will be helpful in building a complete case of how the network acts under the different tests.

9 Design

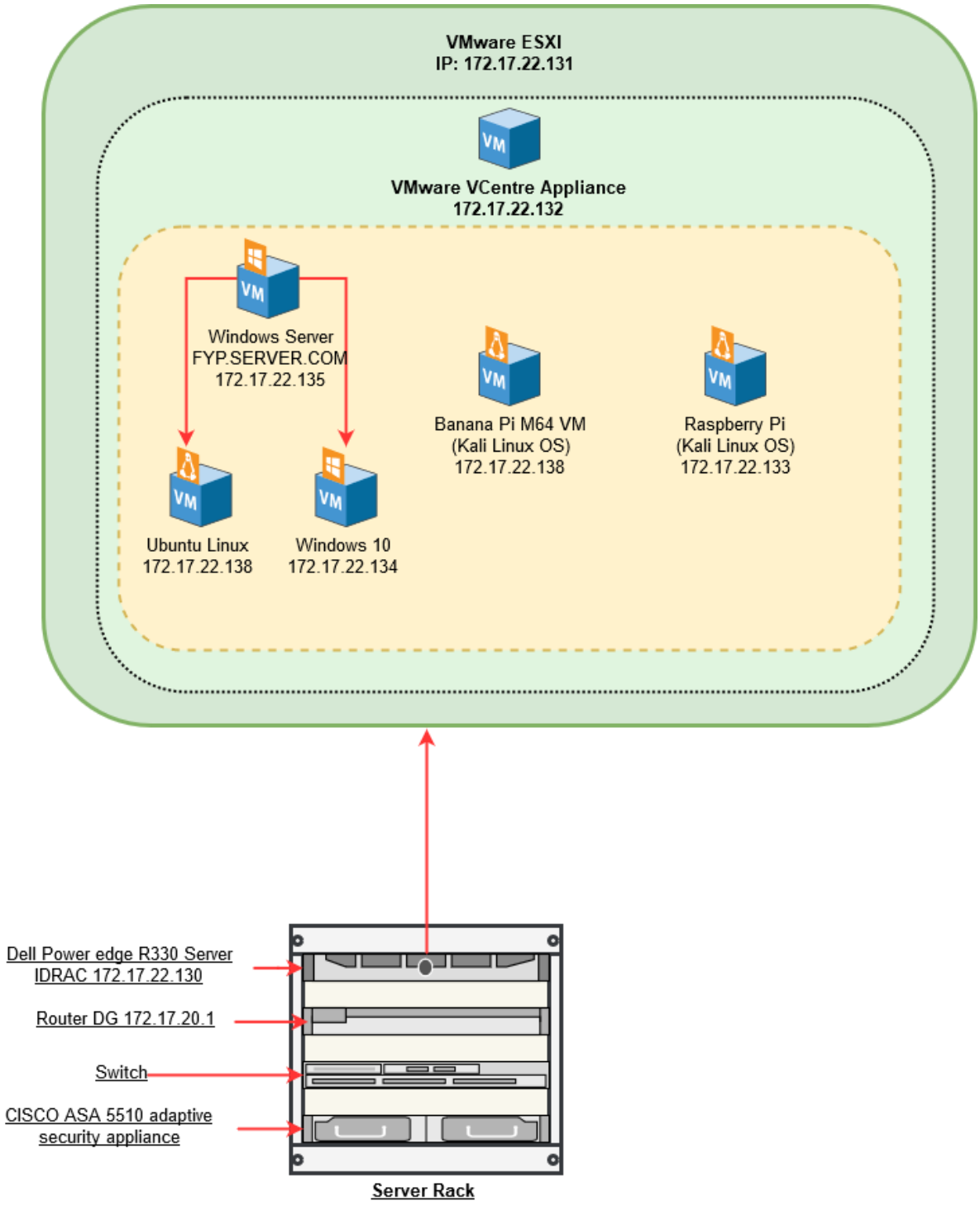


Figure 6 project environment design

10 VMware ESXi

VMware ESXI is a bare metal hypervisor which means it runs directly over a physical server. For this environment VMware 6.5 Hypervisor will be used to virtualize the environment and it will be installed in Dell PowerEdge r330 physical server.

11 VMware vSphere vCentre

VMware vSphere is a collection of virtualized application that includes ESXi and vCenter Server. The vCenter server gathers all the resources and unifies them as a single resource that can be shared among virtual machines in the whole data centres. Also, the VMware vSphere comes with a web client which can be used by the user to modify and interact with different resources and virtual machines.

12 VMware appliance centre

It is a preconfigured Linux machine that can be installed in the ESXI as a VM. It is a pre-packaged 64-bit application with an embedded PostgreSQL database that supports up to 100 hosts and 3000 virtual machines The VMware server appliance is a collection of all the services, tools and licences in one platform. Also, it provides features such as inventory management, virtual machine migration, high availability, distributed resource scheduling, etc

13 VMware Distributed switches

VMware vSphere Distributed Switch (VDS) provides a centralized interface from which you can configure, monitor and administer virtual machine access switching for the entire data centre (VMware, 2019). The VDS simplifies the network monitoring and configuration which enhances the network troubleshooting. For this environment the default VDS will be used as there is no need to fix or add any new VDS.

14 VMware datacentre

The VMware datacentre will be used to store the ISO images of operating systems that will be used to run the virtual machines. Also, it will store VMs virtual hard drive and all the data created by the VMs.

15 IP addressing

For the IP address allocation, class B IP addresses will be allocated to the VMs. For this environment an IP range is given from 172.17.22.130 to 172.17.22.139 with netmask 172.17.20.1 and a netmask of 255.255.252.0.

Device	IPV4 Address
Range	172.17.22.130-172.17.22.139
Netmask	255.255.252.0
Default gateway	172.17.20.1
DNS	8.8.8.8, 8.8.4.4
Dell IDRAC	172.17.22.130
ESXi	172.17.22.131
vCenter Server Appliance	172.17.22.132
Kali Linux 1	172.17.22.133
Windows 10	172.17.22.134
Windows server	172.17.22.135
Windows 10	172.17.22.136
Kali Linux 2	172.17.22.137
Ubuntu	172.17.22.138

Table 4 IP addressing design

16 penetration testing VMs

Penetration testing, also called pen testing or ethical hacking, is the practice of testing a computer system, network or web application to find security vulnerabilities that an attacker could exploit. Penetration testing can be automated with software applications or performed manually.

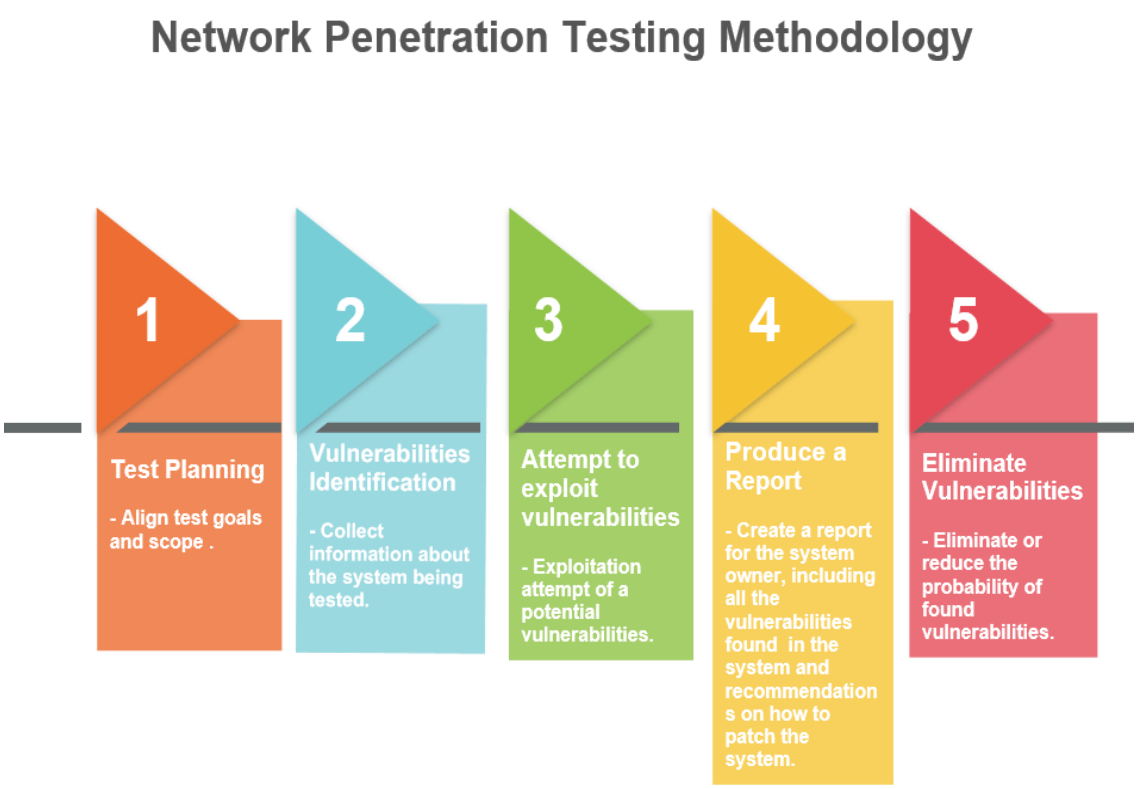


Figure 7 Network pen-testing methodology

These VMs will run Kali Linux as a main operating system. These VMs will be created and configured to mimic the physical portable devices computing power when it comes to storage and CPU power and network limitation.

16.1 Internal pen testing VMs

A Penetration Test mimics the actions of an actual attacker exploiting weaknesses in network security without the usual dangers. This test examines internal IT systems for any weakness that could be used to disrupt the confidentiality, availability or integrity of the network, thereby allowing the organisation to address each weakness. These VMs will contain multiple third-party tools to discover vulnerabilities. The job of these VMs will be identifying live systems, open ports, filtered ports, services running on the ports, mapping network routes and identifying weak firewall rules, etc….

The operating system that will be used in the VM is Kali Linux, some internal scanning tools that might be used:

1- Nmap
2- Superscan

3- Hping
4- Httprint
5- POF
6- Queso
7- Xprobe2
8- Amap

17 Intrusion detection system VM

An intrusion detection system (IDS) monitors network traffic and monitors for suspicious activity and alerts the system or network administrator. These VMs will be using windows server 2016 OS and Ubuntu OS. These VMs will be created and configured to mimic the physical portable devices computing power when it comes to storage and CPU power and network limitation.

18 Intrusion detection software

18.1 Snort

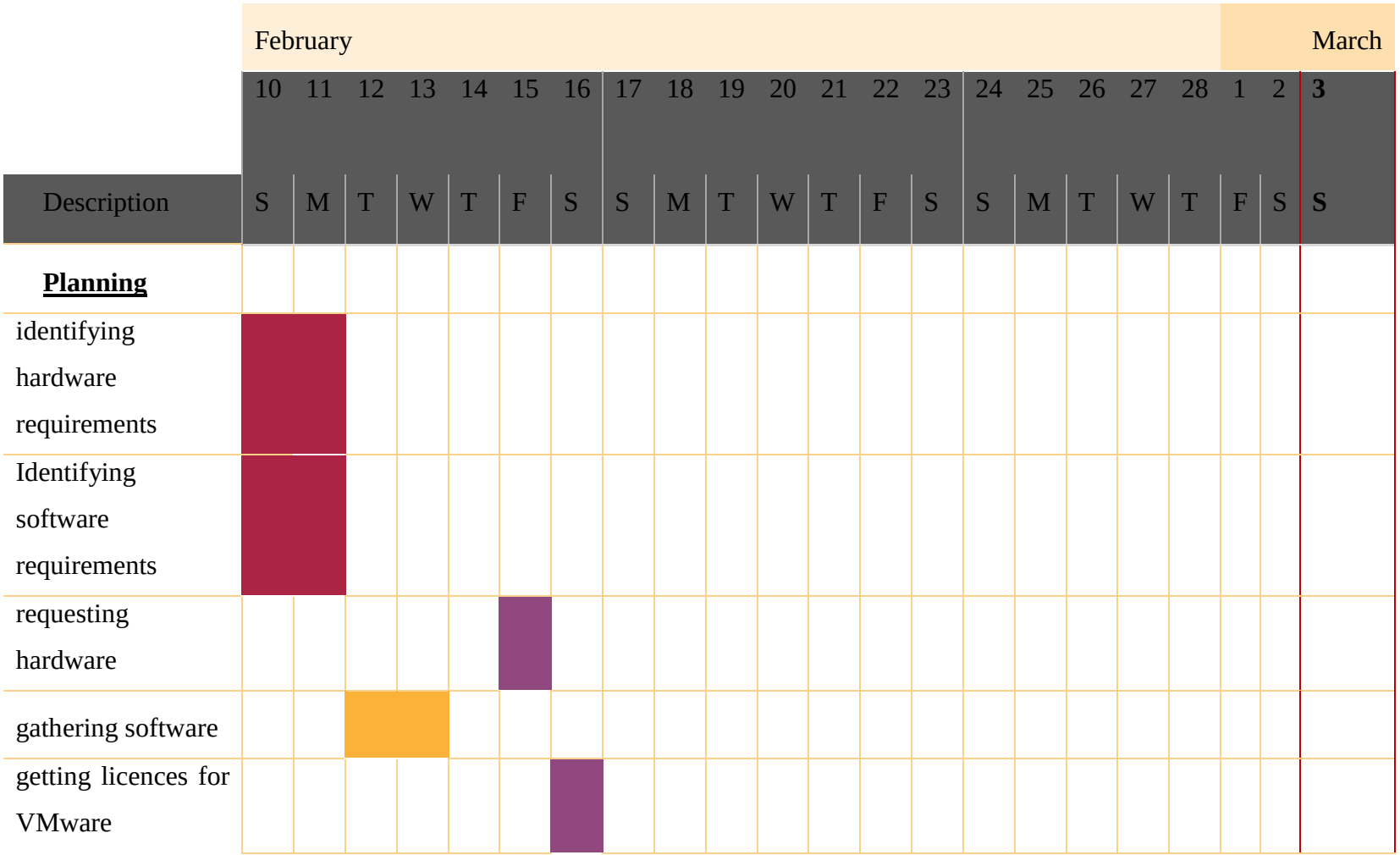
Snort for Windows is an open source network intrusion software that can perform real-time traffic analysis and packet logging on IP networks. The software can perform protocol analysis, content searching/matching and can be used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts and much more. However, you decide to use the software, you will find out that it is a robust tool for gathering and for analysing network traffic. With its add-ons the software can perform just as good as the most commercial IDS products.

18.2 The Bro Network Security Monitor:

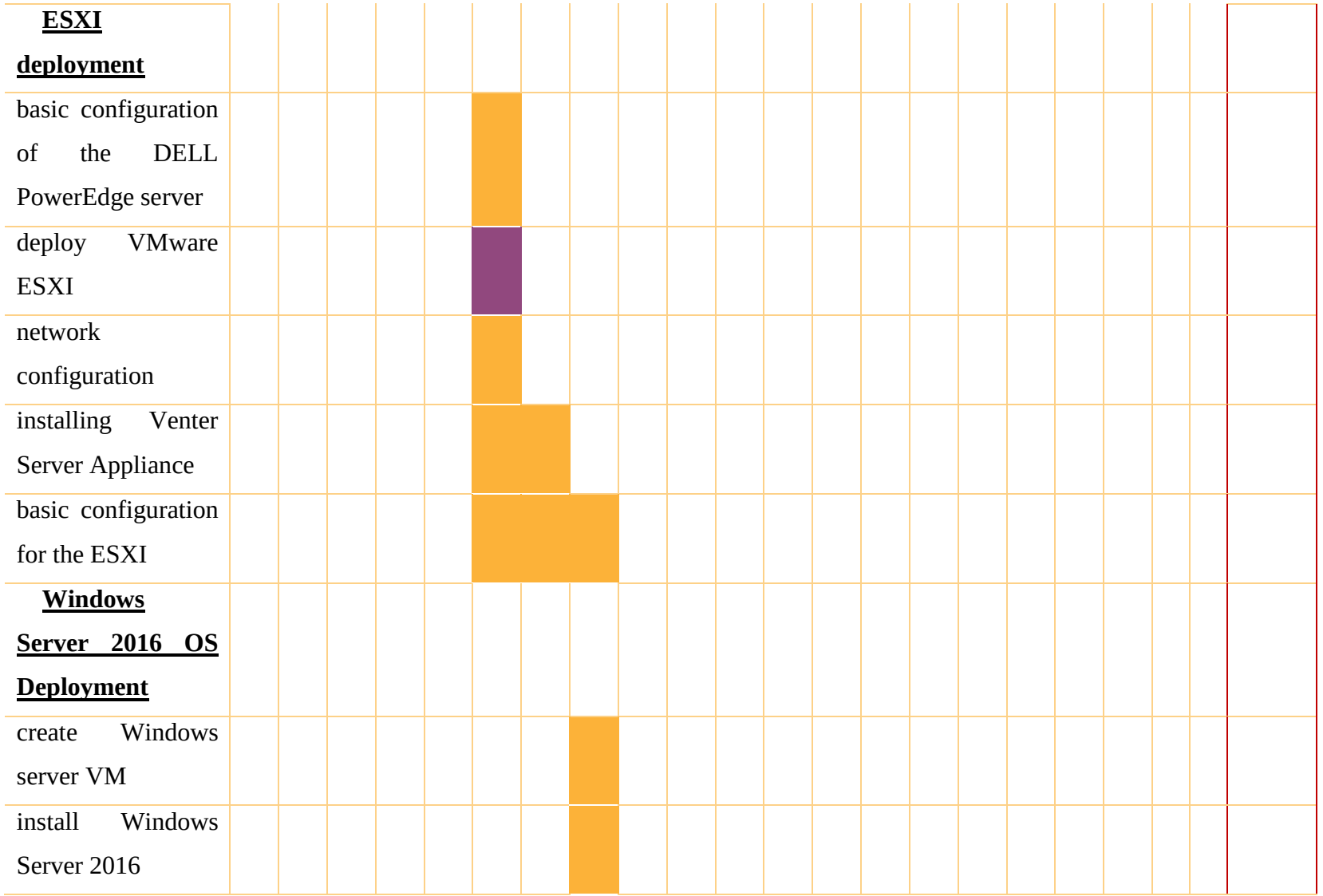
This is a powerful network analysis framework that is very different from the typical IDS you may have known until now. Bro's domain-specific scripting language will enable site-specific monitoring policies. The software targets especially high-performance networks, and it is used operationally at a variety of large sites. The program comes packed with analysers for lots of protocols, and it enables high-level semantic analysis in the application layer. It also keeps great application layer state about the network that it monitors.

Implementation Chapter

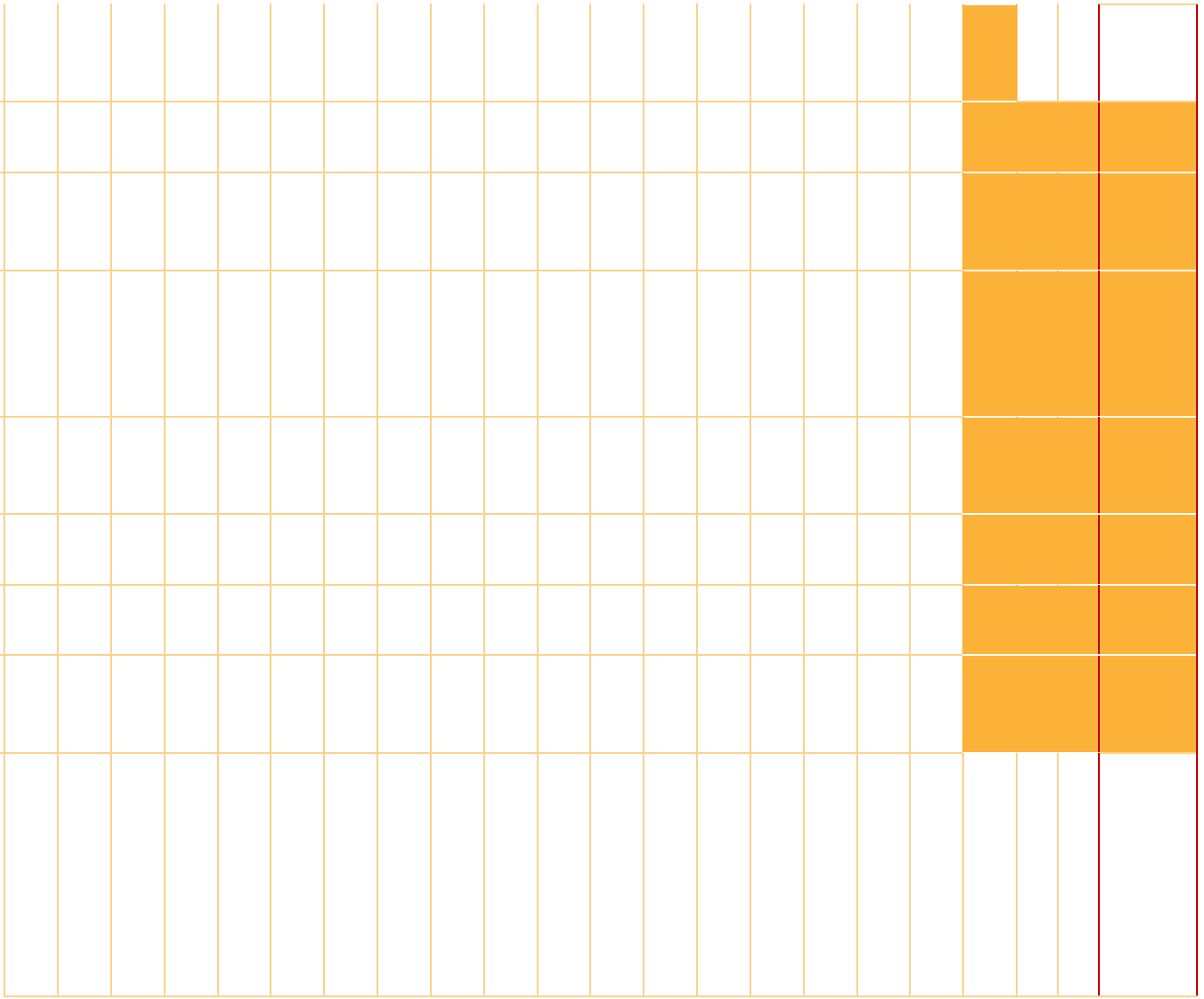
19 Project implementation Gantt chart

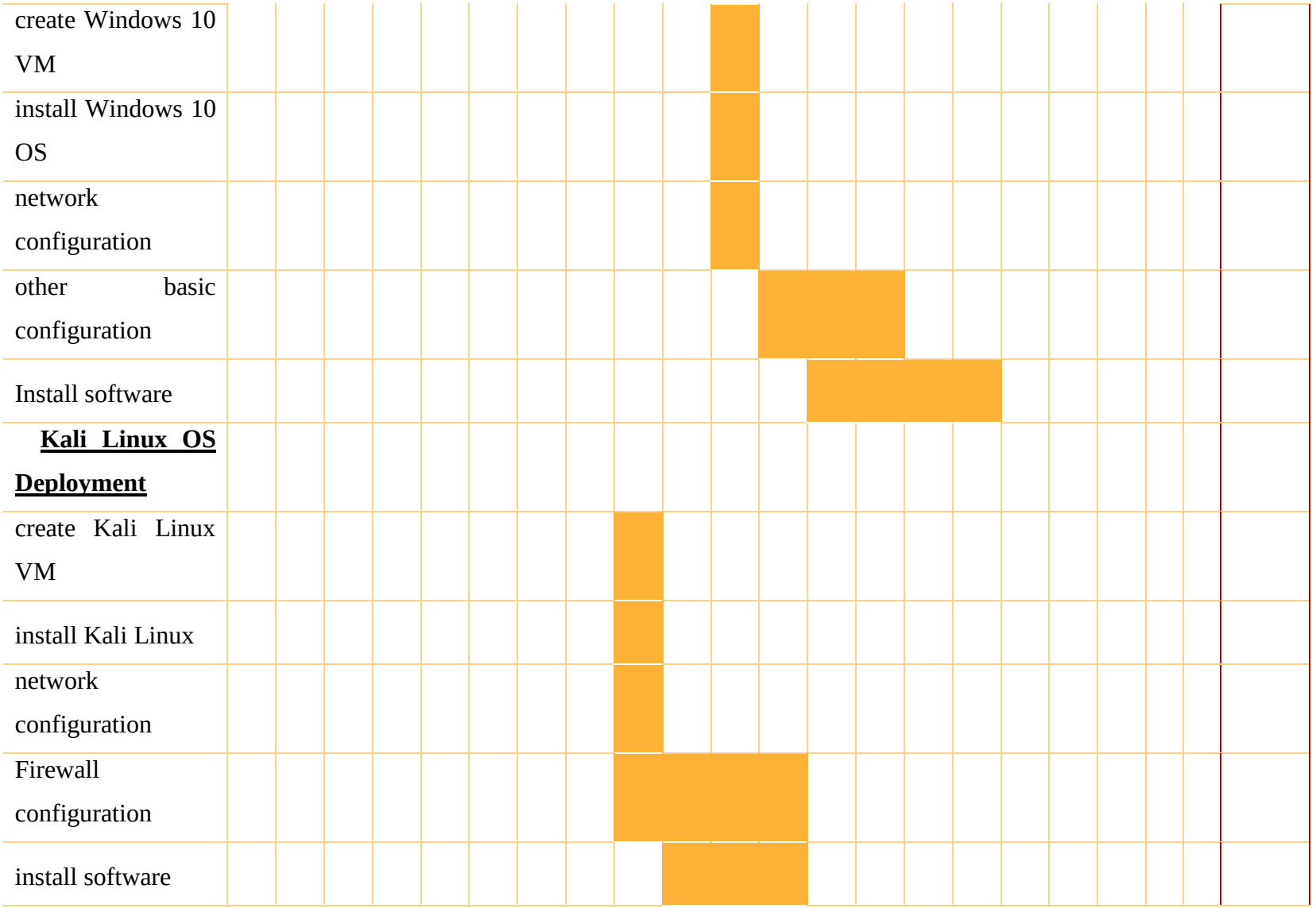


ESXI deployment	
basic configuration of the DELL PowerEdge server	
deploy VMware ESXI	
network configuration	
installing Venter Server Appliance	
basic configuration for the ESXI	
Windows Server 2016 OS Deployment	
create Windows server VM	
install Windows Server 2016	



network configuration	
DNS configuration	
Firewall configuration	
users, groups and active directory configuration	
DHCP configuration	
Domain controller	
install software	
complete the hardening checklist	
Windows 10 OS Deployment	




create Windows 10 VM
install Windows 10 OS
network configuration
other basic configuration
Install software
Kali Linux OS Deployment
create Kali Linux VM
install Kali Linux
network configuration
Firewall configuration
install software

20 abstract

this is the implementation of a virtualize environment, to study the possibility of using Intrusion detection systems and Penetration Testing software in Portable/Micro computing devices and then study the performance of these machines and compare them to the security appliances being used now days. Also, it is going to help in investigating the idea of the possibility of creating neural network with these devices and implementing them in a pre-existing network infrastructure.

21 List of needed hardware

1. USB Stick 32GB
2. Switch
3. Server
4. Workstation

22 Used Hardware Specification

1. HP V1910-24G Switch JE006A
2. Netgear Model FS728TP Switch
3. Cisco ASA 5510 Series adaptive security appliance
4. Dell PowerEdge R330 server
    a. 16GB RAM
    b. Sata6 400GB SSD x2 Raid
5. Cisco Router
6. Workstation
    a. 8 GB RAM minimum
    b. 500 GB HDD minimum
    c. Network connection

23 List of needed software

1. Kali Linux 64bit OS
2. Windows server 2016
3. Windows 10/8/8.1
4. Ubuntu.
5. ESXI 6.5/6.7
6. vCenter Server Appliance
7. TeamViewer

8. Firefox browser
9. Nmap (Linux)
10. SuperScan (Linux)
11. Hping (Linux)
12. Spiceworks (or any network monitoring tool)
13. POF (Linux)
14. Xprobe2 (Linux)
15. Nessus (Linux)
16. Retin-a (Linux)
17. GEI LANguard
18. ISS scanner (Linux)
19. SARA (Linux)
20. Shadow security scanner (Windows)
21. Suricata (windows)
22. Snort (windows & Linux)
23. The Bro Network Security Monitor (windows)

24 Install ESXI on Dell PowerEdge Server

1. Download VMware vSphere Hypervisor 6.5
2. Copy the ISO image of VMware vSphere Hypervisor 6.5 into a USB stick.
3. Make sure to plug in the USB stick in the server before powering on.
4. Go to the BIOS Boot Manager by clicking F11.
5. From the selection menu, choose the USB Stick. it should bring up an ESXi standard boot menu and the standard installer
6. Press ENTER to run the installer.
7. Once loaded, you should see the **Welcome to VMware ESXi X.X.X Installer**. To continue with the installation, go ahead and press Enter.
8. Finish assigning and selecting the appropriate hard disk for the installation.
9. Configure the Network IP address, Default gateway and

25 Installing vCenter Server Appliance

1. Download vCenter Server Appliance 6.5/6.7.
2. Mount the iso image then locate and run vcsa-ui-installer\win32**installer.exe.**
3. Once it finishes loading click install.

4. Choose the deploy option and agree to the terms of service,
5. select to embed the Platform Services Controller (PSC) with the vCenter Server
6. Then specify the ESXi or the vCenter Server where the appliance will be deployed.
7. Specify the VM Name and the root password for the VCSA.
8. Next choose the appliance size, select tiny.
9. Choose a data store where the VM will be deployed and click on next.
10. In the next screen, specify the network configuration of the VCSA.
11. To run the appliance deployment, click on finish.
12. To check if it was successful or not, connect to the ESXi from the web interface, you can see that the VM is well deployed.
13. Step 2, we will configure the appliance.
14. Next specify some NTP server to synchronize the time. Make sure to enable SSH.
15. In the next screen, provide SSO information to manage your vSphere infrastructure.
16. Next you can accept to join the VMWare's Customer Experience Improvement Program (CEIP) or not.
17. To finish, click on finish to run the configuration.
18. Once the configuration is finished, the installer will provide you with the VMware VSphere ESXi web client address and IP address.
19. You can now connect to the vSphere Web Client. Using the link provided at the end of the installation or the IP address.

26 Uploading to the data store

1. Have all the needed file needed for the upload ready.
2. From the VMware ESXi web page, select Data store.
3. Choose database Browser and a new window will open.
4. Select upload and a new window will open.
5. Navigate to the files needed for the upload, then highlight and finish.
6. A loading bar will appear to show the process of the upload.

27 Creating a vSwitch in VMware ESXi 6.5 web

1. Log in to the VMware ESXi 6.5 server.
2. Go to Networking tab left side then Virtual Switch → click the Add Standard Virtual switch to add new vSwitch.

3. Type vSwitch name in the textbox, type MTU speed in the second Textbox, third select the mode of vSwitch and protocol which do you allow.
4. Click finish and the vSwitch will be deployed.

28 Create VLANs

VlANs will be used for management and to separate some VMs.

1. Browse to a distributed switch in the vSphere Web Client navigator.
2. Click the Manage tab and click Settings.
3. Select Private VLAN and click Edit.
4. Click Add to add a Primary VLAN ID to the list.
5. Click up and down arrows to select a primary private VLAN ID.
6. Click the plus sign (+) next to the Primary VLAN ID to add it to the list. The primary private VLAN also appears under Secondary Private VLAN ID.
7. To add a secondary VLAN, click Add under the Secondary VLAN list, and click the up and down arrows to enter the number for the secondary VLAN.
8. Click the plus sign (+) next to the Secondary VLAN ID to add it to the list.
9. In the Secondary VLAN type column, click into the column to activate a drop-down menu. Select either Isolated or Community for the VLAN type.
10. Click OK.

29 Creating Linux VMs

1. Login to Virtual Center using Web Client or vSphere Client.
2. Select ESXi Host and do a Right Click.
3. Click on New Virtual Machine. It will open New Virtual Machine Wizard.
4. Choose "**Create a new virtual machine**" and Click on **Next**.
5. Give the VM a name, choose the compatibility 6.5 then select the suitable guest OS family (Linux) and the version of the operating system.
6. next, select the datastore for storage and installation.
7. Next, configure the needed RAM, CPU and attach the ISO Image of the needed OS to the VM.
8. Click finish, then it will automatically deploy in the server.

30 Creating windows VMs

1. The process is very similar to creating Linux VMs.
2. Login to Virtual Center using Web Client or vSphere Client.

3. Select ESXi Host and do a Right Click.
4. Click on New Virtual Machine. It will open New Virtual Machine Wizard.
5. Choose "**Create a new virtual machine**" and Click on **Next**.
6. Give the VM a name, choose the compatibility 6.5 then select the suitable guest OS family (Windows) and the version of the operating system.
7. Next, select the data store for storage and installation.
8. Next, configure the needed RAM, CPU and attach the ISO Image of the needed OS to the VM.
9. Click finish, then it will automatically deploy in the server.

31 Deploying Windows 10 operating system

1. Login into Virtual Center using Web Client or vSphere Client.
2. After creating the VM, power on the virtual machine and a new window will popup
3. On the Windows Setup screen, select your language, time and currency format, and keyboard layout. Click **Next** to continue.
4. When you reach the installer screen, select (**Install Now)** and follow the instructions provided by the installer to deploy Windows 10 on the VM.
5. When you see the Activate Windows screen, you'll need to either enter a key or skip it.
6. When you reach the (**Which type of installation do you want?)** screen, click (**Custom**) to perform a clean installation and remove everything on the VM.
7. On the next screen, select the hard drive you want to install Windows on and erase it.
8. When you're done erasing partitions, you should have a big block of (**Unallocated Space**). Select that, click (**New**), and once it's formatted your drive, click **(next**).
9. Windows 10 will install itself and may restart a few times during this process.
10. When it's done, you will be asked to adjust various settings regarding the user experience.
11. Once it is done, you can log in into the Operating system and use it.

32 Deploying Windows Server 2016 operating system

1. Login into Virtual Center using Web Client or vSphere Client.

2. After creating the VM, power on the virtual machine and a new window will popup
3. On the Windows Setup screen, select your language, time and currency format, and keyboard layout. Click **Next** to continue.
4. When you reach the installer screen, select (**Install Now)** and follow the instructions provided by the installer to deploy Windows server 2016 on the VM
5. Next, you must choose the version of windows server 2016 to be installed. Select Server 2016 Standard with GUI.

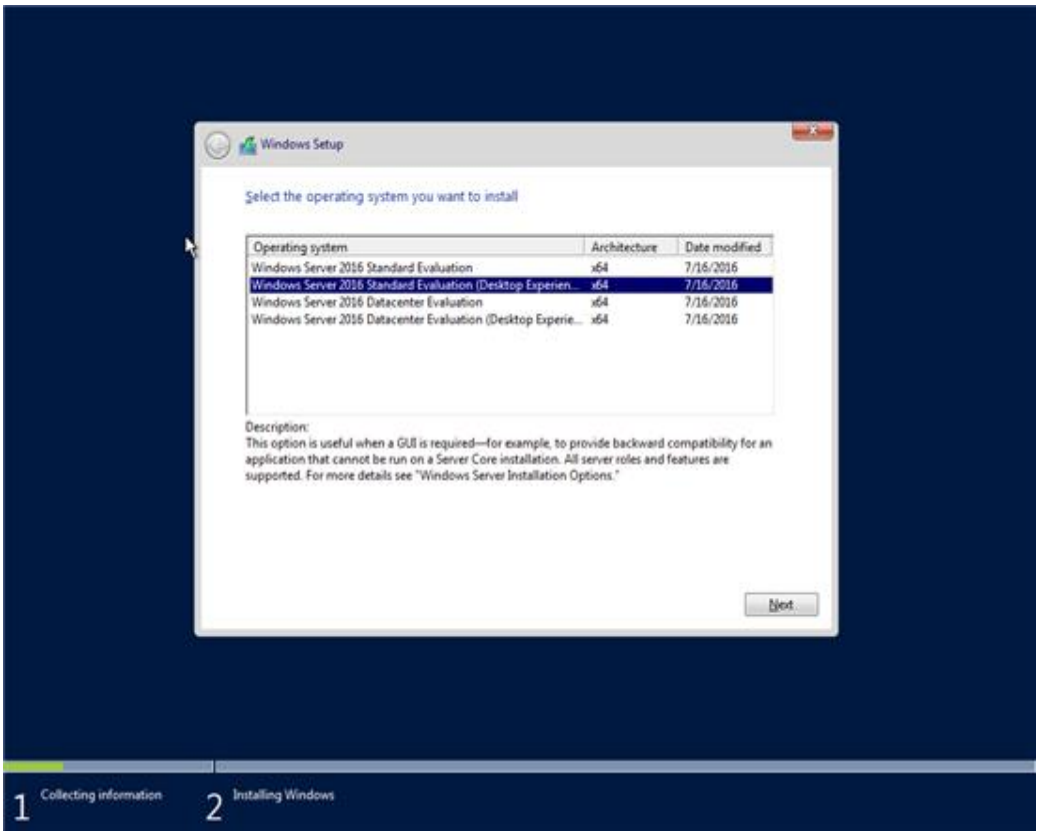


Figure 8 Windows server 2016 versions

6. On the licence terms on this screen, select (**I accept Licence Terms**) then click (**Next**) to continue.
7. When you reach the (Which type of installation do you want?) screen, click (Custom) to perform a clean installation and remove everything on the VM.
8. On the next screen, select the hard drive you want to install Windows on and erase it.
9. When you're done erasing partitions, you should have a big block of (**Unallocated Space**). Select that, click (**New**), and once it's formatted your drive, click **(next**).

10. Windows server 2016 will install itself and may restart a few times during this process.
11. When it’s done, you will be asked to adjust various settings regarding the user experience.
12. Once it is done, you can log in into the Operating system and use it.

33 Deploying Kali Linux Operating System

1. Login into Virtual Center using Web Client or vSphere Client.
2. After creating the VM, power on the virtual machine and a new window will popup.
3. When the installer lunches, highlight Graphical Interface and click ENTER.
4. The next couple of screens will ask you to select local information such as your preferred language, your country location, and keyboard layout.
5. Once through the local information, the loader will automatically install some additional components and configure your network related settings. Then the installer will prompt for a hostname and domain for this installation. Provide appropriate information for the environment and continue installing.
6. Set a password for your Kali Linux machine and hit continue.
7. After the password is set, the installer will prompt you to set the time zone and then pauses at the disk partitioning. The installer will now provide you four choices about the partitions of the disk.
8. Choose guided- Use Entire Disk and hit continue
9. Confirm all changes to be made to the disk on the host machine. Be aware that if you continue it will erase data on the disk.
10. Once you confirm the partition changes, the installer will run through the process of installing the files. Let it install the system automatically, which may take a round 5-15 minutes.
11. Once the necessary files are installed, the system will ask you if you want to set up a network mirror to obtain future pieces of software and updates. Click **yes** to enable this functionality to use the Kali repositories.
12. Next, you will be asked to install the GRUB boot loader. Select **yes** and pick the device to write the necessary boot loader information to the hard drive which is required to boot Kali.

13. Once the installer finishes installing GRUB to the disk, click on **Continue** to finish the installation; it will install some final stage files.
14. After it finishes installing the files, you will be able to log in and use the operating system normally.

34 Creating Snapshots (VM backup)

1. Right-click the virtual machine the inventory and select Snapshots → Take Snapshot.
    a. To locate a virtual machine, select a datacenter, folder, cluster, resource pool, host, or vApp.
    b. Click the VMs tab and click Virtual Machines.
2. Type a name for the snapshot.
3. (Optional) Type a description for the snapshot.
4. (Optional) Select the Snapshot the virtual machine's memory check box to capture the memory of the virtual machine.
5. (Optional) Deselect Snapshot the virtual machine's memory and select the Quiesce guest file system (Needs VMware Tools installed) check box to pause running processes on the guest operating system so that file system contents are in a known consistent state when you take the snapshot.
6. Quitesce the virtual machine files only when the virtual machine is powered on and you do not want to capture the virtual machine's memory.
7. Click OK.

35 IPV4 addressing

Device	IPV4 Address
Dell Idrac	172.17.22.130
ESXi	172.17.22.131
vCenter Server Appliance	172.17.22.132
Kali Linux 1	172.17.22.133
Windows 10	172.17.22.134
Windows server	172.17.22.135
Windows 10	172.17.22.136
Kali Linux 2	172.17.22.137
Ubuntu	172.17.22.138

Table 5 IPv4 addresses

36 Configuring static IPV4 in kali Linux using GUI.

1. Click on the arrow in the upper right corner of the screen, in the menu that appears, click on ‘Wired Connected’, and then on ‘Wired Settings’.
2. A new window will open, in it click on the gear icon.
3. Another window will open, in which go to the IPv4 tab.
4. Select Manual.
5. Enter the desired static IP in the Addresses field, it should match your network.
6. Click apply, then ok.

37 Configuring static IPV4 in kali Linux using command line

1. Open the /etc/network/interfaces file with any text editor using this command.
   sudo vim /etc/network/interfaces
2. Only four lines should be added to this file:
   auto eth0
   iface eth0 inet static
   address 17.17.22.133/22
   gateway 172.17.20.1
3. For the changes to take effect, issue the command:

sudo systemctl restart networking.service

4. Check the IP address by issuing the command:
   Ip a

38 Configure static IPV4 in Windows 10/8/8.1

1. Open Control Panel.
2. Click on Network and Internet.
3. Click on Network and Sharing Center.
4. On the left pane, click the Change adapter settings link.
5. Right-click the network adapter and select Properties.
6. Select the Internet Protocol Version 4 (TCP/IPv4) option.
7. Click the Properties button and a new window will open.
8. Populate all the fields with the appropriate IP addressing,
9. Click ok then apply and close.

39 Add Windows Server 2016 Roles and Features

1. Select Start → Server Manager.
2. In Server Manager, select Manage → Add Roles and Features.
3. In the Add Roles and Features wizard, click Next until the Server Roles page appears.
4. In the Server Roles page, select the roles you need by checking the check box.
5. Click Next.
6. Click Next and in the Confirmation page, click Install.
7. When the installation is finished, click Close to close the Add Roles and Features wizard.

40 Windows Server 2016 Domain controller

1. Open server manager
2. Navigate to Manage → Add Roles and Features
3. Select Installation Type → Role-based or feature-based installation → Next.
4. Select Server Selection → Select a Server from the server pool → Your server → Next.
5. At the Server Roles screen choose Active Directory Domain Services → Next.
6. You will now be asked to add several other features (the RSAT tools). Click on Add Features to confirm and then click Next.
7. Click Next to leave the Features screen.
8. Click Next to leave the AD DS Screen.
9. On the Confirmation screen choose whether to reboot the server when the Roles and Features are installed and click Next to proceed with the install.
10. Once the installation is complete and you restart the server, launch Server Manager again so that we can finish promoting the machine to a DC.
11. You will see a warning icon in front of Manage, click the icon and then click "Promote this server to a domain controller".
12. You will now be asked to add several other features (the RSAT tools). Click on Add Features to confirm and then click Next.
13. Click Next to leave the Features screen.
14. Click Next to leave the AD DS Screen.
15. On the Confirmation screen choose whether to reboot the server when the Roles and Features are installed and click Next to proceed with the install.

16. Once the installation is complete and you restart the server, launch Server Manager again so that we can finish promoting the machine to a DC.
17. You will see a warning icon in front of Manage, click the icon and then click "Promote this server to a domain controller"
18. At the Deployment Configuration screen select "Add a domain controller to an existing domain," select the domain, set the credentials to use, and then click Next.
19. Choose the appropriate options, for this example we are going to select Domain Name System (DNS) server and Global Catalog (GC) so that our DC acts as a proper secondary DC and DNS server for our domain. Click Next.
20. If you receive the warning "A delegation for this DNS server cannot be created because the authoritative parent zone cannot be found…" you can safely ignore the message and proceed. This error may pop up if you're not using a parent zone and is a non-issue if you don't need name resolution from outside the domain.
21. On the additional options screen choose which domain controllers you want to replicate from. Since I'm not decommissioning any servers soon I chose Any Domain Controller. Click Next.
22. At the Paths screen leave the defaults (unless you have a good reason not to) and click Next.
23. At the Review Options screen verify everything looks good, optionally view the PowerShell script, and click Next.
24. The prerequisite check will run. Once it's complete click Install.
25. Once the install completes your machine should warn you that it's about to restart. Let it restart and you'll be ready to start playing with your new DC.

41 Windows Server 2016 Groups and policies

1. open Server Manager console → Tools → Click on Group Policy management.
2. We want to apply policy on a User Group, Right -click on the user group → and select Create a GPO in this domain, and Link here...
3. Then, you must enter the name of the new GPO, enter any name for GPO in the Name field → when finish click on OK.
4. Right Click on the Group Policy Object name and then click edit.

5. Here you must choose the policy which you want to apply. If you want to apply the policy to the users, then go to User Configuration and if you want to apply the policy on the computer then choose Computer Configurations.
6. A new window of Group Policy Management editor will open. Expand User Configuration → Policies → Administrative Templates → Control Panel → Double Click on Display.
7. On the right side. Double Click on Disable the Display Control Panel. Here Select Enabled and Click on OK.
8. Then apply and OK

42 Windows Server 2016 Users

1. Choose Start → Administrative Tools → Active Directory Users and Computers.
2. Right-click the domain that you want to add the user to and then choose New → User from the contextual menu.
3. Enter the user's first name, middle initial, and last name.
4. Change the Full Name field if you want it to appear different from what the wizard proposes.
5. Enter the user logon name. This name must be unique within the domain.
6. Click next and add a password.
7. Specify the password options that you want to apply.
8. Click next and verify the information then click on finish

43 Windows Server 2016 Firewall configuration

1. Start system and login (with administrator rights user).
2. Open Control Panel. Click on Windows Firewall.
3. After that click on Advanced Settings.
4. Windows Firewall console open. If you click on Properties (right side) – you can disable firewall for all networks. You are going to Inbound Rules (left side) for rule creation.
5. Click on New Rule.
6. Rule creation process begin, choose the appropriate settings according to the security rule you want to create by filling the boxes.
7. Once you are done with the settings, click finish.

44 Windows Server 2016 DNS configuration

1. Open the server manager dashboard.
2. Click on Add roles and features.
3. Read the pre-requirements and click Next.
4. Choose Role-based or feature-based installation and click Next.
5. Choose destination server for DNS role and click Next.
6. Choose DNS server from server roles. As soon as you choose the role, a new window will pop up. Click Add features.
7. Keep clicking Next through rest of pages and complete the installation process

45 Creating Forward Lookup Zone

1. Open server manager dashboard.
2. Click Tools → DNS.
3. In DNS manager console, expand **DNS server**. Right-click **Forward Lookup Zones**.
4. Click **new zone**.
5. Click **Next**.
6. Choose **primary zone** and uncheck **Store the zone in AD**.
7. Provide the zone name and click Next.
8. Choose **Create a new file with this ....** and click **Next**.
9. Choose **Do not allow dynamic updates**.
10. Click next, then Finish.

46 Windows Server 2016 DHCP setup

1. Open server manager
2. Navigate to Manage → Add Roles and Features
3. Select Installation Type → Role-based or feature-based installation → Next.
4. Select Server Selection → Select a Server from the server pool → Your server → Next.
5. At the Server Roles screen choose **DHCP Server** → Next.
6. Click next through the next screens and then finish.
7. Once done with the installation, restart the machine.

47 Windows Server 2016 DHCP configuration

1. On the Server Manager console, click Tools, and then click DHCP.

2. On the DHCP console, expand your server name FYP.Server
3. Select and right-click your server name and then select Authorize to authorize your DHCP server.
4. Refresh the DHCP console to enforce the changes.
5. Now, create a new DHCP scope to specify the IP address ranges for your DHCP server. To do so, select and right-click IPv4 and then select New Scope.
6. On the welcome page of the New Scope Wizard, click Next.
7. On the Scope Name page, specify a DHCP scope name as shown in the following figure, and then click Next.
8. On the IP Address Range page, specify the start and end IP addresses from which the DHCP server will allocate the IP addresses to the clients
9. On the Add Exclusions and Delay page, exclude the IP addresses that you want to be not distributed by the DHCP server.
10. Click Add, and then click Next. On the Lease Duration page, review the default lease duration limit, and then click Next.
11. On the Configure DHCP Options page, make sure that the Yes, I want to configure these options now radio button is selected and then click Next.
12. On the Router (Default Gateway) page, in the IP address text box, type the address of your router.
13. Click Add, and then click Next. On the Domain Name and DNS Servers page, make sure that your DNS server IP address is already populated. click Next.
14. On the next screen leave it empty and click next.
15. On the Activate Scope page, make sure that the Yes, I want to activate this scope now radio button is selected. Click Next and complete the wizard.
16. Refresh the DHCP console. Make sure that the IPv4 node is marked with the green color.

48 Windows Server 2016 Hardening checklist

1. The hardening checklists are based on the comprehensive checklists produced by the Centre for Internet Security (CIS).
2. Make sure to compete more than 80% before hardening the system

Step	To Do	√
	Preparation and Installation	
1	If machine is a new install, protect it from hostile network traffic, until the operating system is installed and hardened.	√
	Service Packs and Hotfixes	
2	Install the latest service packs and hotfixes from Microsoft.	√
3	Enable automatic notification of patch availability.	√
	User Account Policies	
4	Set minimum password length.	√
5	Enable password complexity requirements.	√
6	Do not store passwords using reversible encryption. (Default)	√
7	Configure account lockout policy.	√
	User Rights Assignment	
8	Restrict the ability to access this computer from the network to Administrators and Authenticated Users.	√
9	Do not grant any users the 'act as part of the operating system' right. (Default)	√
10	Restrict local logon access to Administrators.	√
11	Deny guest accounts the ability to logon as a service, a batch job, locally, or via RDP.	√
	Security Settings	
12	Place a warning banner in the Message Text for users attempting to log on.	√
13	Disallow users from creating and logging in with Microsoft accounts.	√
14	Disable the guest account. (Default)	√
15	Require Ctrl+Alt+Del for interactive logins. (Default)	√
16	Configure machine inactivity limit to protect idle interactive sessions.	√
17	Configure Microsoft Network Client to always digitally sign communications.	√

18	Configure Microsoft Network Client to digitally sign communications if server agrees. (Default)	√
19	Disable the sending of unencrypted passwords to third party SMB servers.	x
20	Configure Microsoft Network Server to always digitally sign communications.	x
21	Configure Microsoft Network Server to digitally sign communications if client agrees.	x
	Network Access Controls	
22	Disable anonymous SID/Name translation. (Default)	√
23	Do not allow anonymous enumeration of SAM accounts. (Default)	√
24	Do not allow anonymous enumeration of SAM accounts and shares.	√
25	Do not allow everyone permissions to apply to anonymous users. (Default)	√
26	Do not allow any named pipes to be accessed anonymously.	√
27	Restrict anonymous access to named pipes and shares. (Default)	√
28	Do not allow any shares to be accessed anonymously.	√
29	Require the "Classic" sharing and security model for local accounts. (Default)	√
	Network Security Settings	√
30	Allow Local System to use computer identity for NTLM.	√
31	Disable Local System NULL session fallback.	√
32	Configure allowable encryption types for Kerberos.	x
33	Do not store LAN Manager hash values.	x
34	Set LAN Manager authentication level to only allow NTLMv2 and refuse LM and NTLM.	x
35	Enable the Windows Firewall in all profiles (domain, private, public). (Default)	√
36	Configure the Windows Firewall in all profiles to block inbound traffic by default. (Default)	√
37	Configure Windows Firewall to restrict remote access services (VNC, RDP, etc.)	√
38	Configure Windows Firewall to restrict remote access services (VNC, RDP, etc.) to the campus VPN.	-
	Active Directory Domain Member Security Settings	

39	Digitally encrypt or sign secure channel data (always). (Default)	√
40	Digitally encrypt secure channel data (when possible). (Default)	√
41	Digitally sign secure channel data (when possible). (Default)	√
42	Require strong (Windows 2000 or later) session keys.	√
43	Configure the number of previous logons to cache.	√
	Audit Policy Settings	
44	Configure Account Logon audit policy.	√
45	Configure Account Management audit policy.	√
46	Configure Logon/Logoff audit policy.	√
47	Configure Policy Change audit policy.	√
48	Configure Privilege Use audit policy.	√
	Event Log Settings	
49	Configure Event Log retention method and size.	x
50	Configure log shipping (e.g. to Splunk).	x
	Linux Subsystem	
51	Configure all Linux elements according to the Linux Hardening Guide, keeping in mind that some elements will require Windows tools (like Windows Firewall vs. iptables)	x
	Additional Security Protection	
52	Disable or uninstall unused services.	√
53	Disable or delete unused users.	√
54	Configure user rights to be as secure as possible: Follow the Principle of Least Privilege	√
55	Ensure all volumes are using the NTFS file system.	√
56	Configure file system permissions.	√
57	Configure registry permissions.	√
58	Disallow remote registry access if not required.	√
	Additional Steps	
59	Set the system date/time	√
60	Install and enable anti-virus software.	√
61	Install and enable anti-spyware software.	√
62	Configure anti-virus software to update daily.	√

63	Configure anti-spyware software to update daily.	√
65	Install software to check the integrity of critical operating system files.	√
66	If RDP is utilized, set RDP connection encryption level to high.	x
67	Unless the server is in the UDC or a managed VM cluster, set a BIOS/firmware password to prevent alterations in system start up settings.	x
68	Do not allow the system to be shut down without having to log on. (Default)	√
69	Configure the device boot order to prevent unauthorized booting from alternate media.	x
70	Configure a screen-saver to lock the console's screen automatically if the host is left unattended.	x

Table 6 Windows Server 2016 Hardening checklist

49 Ubuntu installing the Bro Network Security Monitor.

1. Open a new terminal and make sure you have administrator rights.
2. Install the Dependencies by using the following command:

```
sudo apt-get install cmake make gcc g++ flex bison libpcap-dev libgeoip-dev libssl-dev python-dev zlib1g-dev libmagic-dev swig libgoogle-perftools-dev
```

3. Create the directory where Bro will be installed, and it writes its logs. By using the following command:

   ```
   sudo mkdir -p /nsm/bro
   ```

4. Bro is installed by downloading the current source code and building it with the make command:

   ```
   cd ~
   wget https://www.bro.org/downloads/release/bro-2.4.1.tar.gz
   tar -xvzf bro-2.4.1.tar.gz
   cd bro-2.4.1
   ./configure --prefix=/nsm/bro
   make
   sudo make install
   export PATH=/nsm/bro/bin:$PATH
   ```

50 Windows Server 2016 installing Shadow security scanner

1. Go to the following link and download the leatest version of the software:

http://shadow-security-scanner.freedownloadscenter.com/windows/

2. Once the download is finished, run the installer and follow the instruction provided.
3. When the installation process is complete you should be able to run the software.

51 Windows Server 2016 installing Suricata

1. The instruction provided in the following link must be followed.
2. It's very crucial that the steps must be completed in the order they gave in their website for the software to be deployed in windows
   https://redmine.openinfosecfoundation.org/projects/suricata/wiki/Windows

52 Windows Server 2016 installing Snort

1. Download the software from the link provided:
   https://www.securityarchitecture.com/learning/intrusion-detection-systems-learning-with-snort/getting-and-installing-tools/
2. Follow the instruction provided by the developers in the link above and the software should install without any issue.

53 Join Kali Linux to Windows Server 2016 Domain

1. Open the Add/Remove Software tool.
2. Search for **likewise open**.
3. Mark **likewise-open5**, **likewise-open5-gui**, and **winbind** for installation (the Add/Remove tool will pick up any necessary dependencies for you).
4. Click Apply to install (and Apply to accept any dependencies).
5. Open a new terminal, make sure that you have administrator/root privilege
6. issue the following command:

   sudo domainjoin-cli join DOMAIN_NAME USER

   Where DOMAIN_NAME is the name of the Windows domain you want to join, and USER is the user you authenticate with. You will be prompted for your user password and, upon successful authentication.
7. Now if you want to leave the domain you can do so equally as easy with the command:

sudo domainjoin-cli leave

54 Kali Linux Users and Groups

55 Kali Linux Configuring SSH for Remote Logins

1. Open a new terminal.
2. Type in the following command to run the SSH services manually

   systemctl start ssh
3. Type in the following command to stop the SSH services manually

   systemctl stop ssh
4. alternatively, SSH can be configured to run on boot by suing the following command:

   systemctl enable ssh

56 Install Nmap in Kali Linux using command line

1. open new command line terminal
2. First you will need to obtain the necessary dependencies in order for Nmap to install successfully

   $ sudo apt-get install git wget build-essential checkinstall libpcre3-dev libssl-dev
3. Next, you'll want to check out the source repository from the Nmap GitHub page in order to install Nmap from the latest available version.

   $ git clone https://github.com/nmap/nmap.git
4. Next run the configuration script. If you see any errors or warnings make sure you go back and install the necessary libraries from Step 1.

   $ ./configure
5. Next run makes and then make install to compile and install Nmap on your system.

   $ make && make install
6. Verify installation

   $ nmap -V

57 Install super-scan in Kali Linux

1. Go the website and download the latest version of super scan.

   http://pkg.kali.org/pkg/simple-scan
2. Install the super scan by clicking on install.

58 Install Hping in Kali Linux

1. Open a new command terminal.

2. Make sure to have high privilege for the installation before typing the following command:

   sudo apt-get install hping3

59 Installing Nessus in Kali Linux

1. Nessus is not a free software, so a trail version is being used for this implementation. Go to the following link and follow the signup popup and download the Linux version of the software:

   https://www.tenable.com/try

2. Obtain Nessus and an Activation Code - Once you've signed up for Nessus, you will receive an account on the Tenable Support Portal and an activation code to be used in the installation process.
3. Once you've transferred the appropriate Nessus Debian package to your Kali Linux installation, run the following commands to install and start Nessus

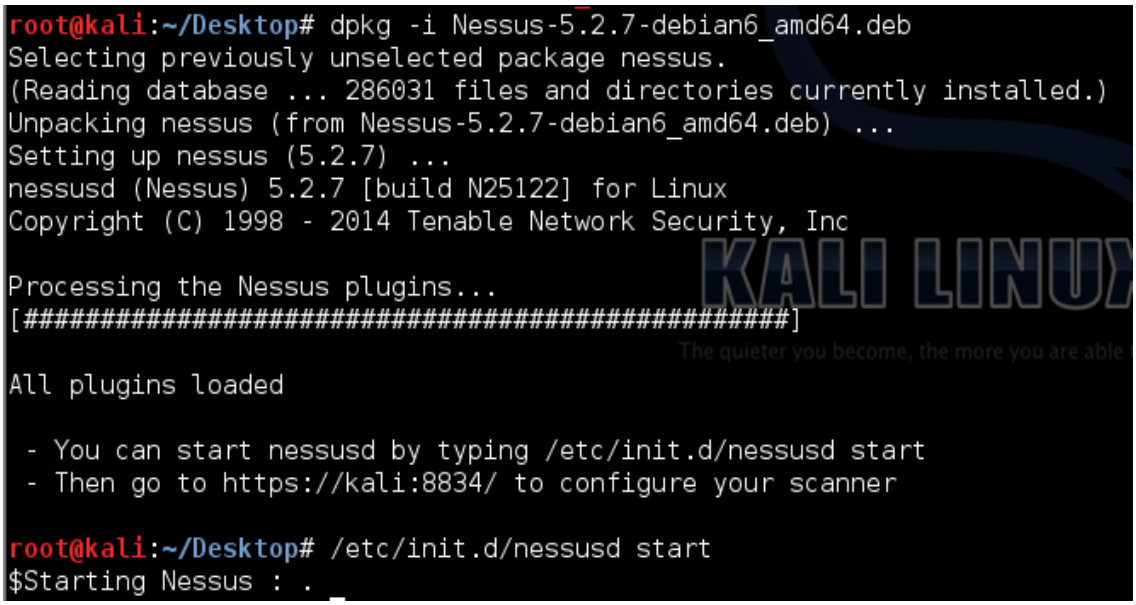


Figure 9 Nessus installation terminal

4. The web interface can be accessed with the browser by making an HTTPS connection to TCP port 8834 (e.g. https://localhost:8834/). I can also access the Nessus Web Interface remotely by using the default IP address assigned to Kali Linux (e.g. https://172.17.22.133:8834/). Make certain that JavaScript is enabled in the browser you are using to manage the Nessus server.

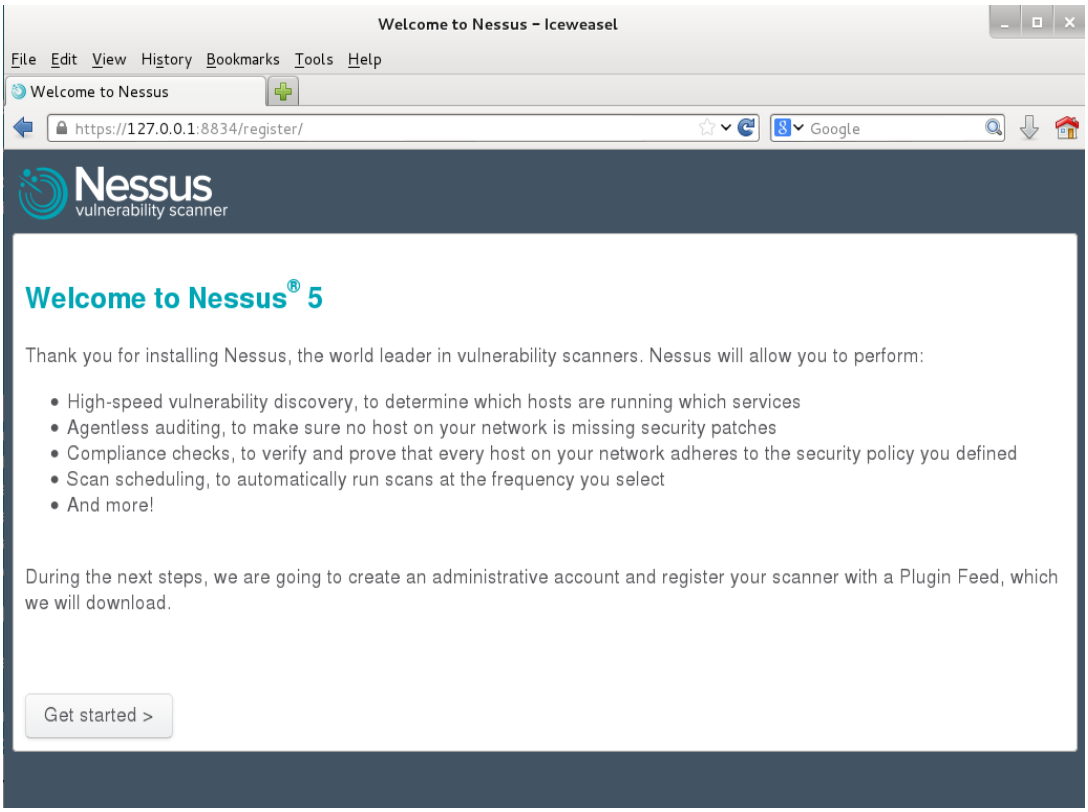


Figure 10 Nessus web manager

60 Installing Xprobe2 in Kali Linux

1. Make sure to have the required repositories, open a new terminal and type in the following command:

   # sudo apt-get install libpcap0.8-dev

2. You will also need g++-4.1 since Xprobe will only compile with version 4.1 of GCC:

   # sudo apt-get install g++-4.1

3. After finishing step 1 and 2, start the installation process of Xporobe2 by running the following lines:

   cd /data/src/

   wget

   http://downloads.sourceforge.net/project/xprobe/xprobe2/Xprobe2%200.3/xprobe2-0.3.tar.gz

```
tar xzvf xprobe2-0.3.tar.gz
cd xprobe2-0.3/
./configure CC=gcc-4.1 CXX=g++-4.1
make
sudo make install
```

61 Install Snort in kali Linux

1. Open a new command line terminal
2. Type in these two commands:

   ```
   # apt-get update
   # apt-get install snort
   ```
3. Verify the Snort Installation

   ```
   # snort –version
   ```
4. Create the following snort.conf and icmp.rules files
   a. Open the configuration file of snort

      ```
      # leafpad /etc/snort/snort.conf
      ```
   b. Check the configuration file and check whether the icmp rules is included or not. If not, include the line below.

      ```
      include /etc/snort/rules/icmp.rules
      ```
   c. Open icmp rules file and include a rule mentioned below

      ```
      # leafpad /etc/snort/rules/icmp.rules
      ```
   d. Include the below mentioned line into icmp.rule file.

      ```
      alert icmp any any -> any any (msg:"ICMP Packet"; sid:477; rev:3;)
      ```

      The above basic rule does an alert when there is an ICMP packet (ping).

5. Following is the structure of the alert:

   <Rule Actions> <Protocol> <Source IP Address> <Source Port> <Direction Operator> <Destination IP Address> <Destination > (rule options)

Rules options	
Structure	example
Rules action	alert
Protocol	ICMP
Source IP address	any
Source port	Any
Direction operator	->
Destination IP address	Any
Destination port	Any
(rules options)	(msg:"ICMP Packet"; sid:477; rev:3;)

Table 7 Snort Rules

6. Execute snort from command line, as mentioned below.

   # snort -c /etc/snort/snort.conf -l /var/log/snort/

62 Accounts, users, privilege and password

OS/System	Privilege	Username/account name	password
Dell Idrac	Administrator	root	Networks4!
ESXI	Administrator	root	Networks4!
Windows server 2016	Administrator	administrator	Networks4!
Windows server 2016	Administrator	User1	Networks4!
Windows server 2016	Low privilege user	User2	Networkss44!
Windows server domain		FYP.server.com	Networks4!
Windows 10	Administrator	Administrator	Networks4!
Windows 10	Administrator	User1	Networks4!
Windows 10	Low privilege user	User2	Networkss44!
Kali Linux 1	Administrator	Administrator	Networks4!
Kali Linux 1	Administrator	User1	Networks4!
Kali Linux 1	Low privilege user	User2	Networkss44!
Kali Linux 2	Administrator	Administrator	Networks4!
Kali Linux 2	Administrator	User1	Networks4!
Kali Linux 2	Low privilege user	User2	Networkss44!

Table 8 Accounts, users, privilege and password

63 Installation status

ID	To do	status	notes
1	Configure IDRAC	●	
2	Install ESXi hypervisor	●	
3	Install VCenter server appliance	●	
4	Configure data store	●	
5	ESXI network configuration	●	
6	Creating Linux VM	●	
7	Creating Windows 10 vm	●	
8	Creating windows server 2016 VM	●	
9	Deploying Kali Linux OS	●	
10	Deploying Windows server 2016 OS	●	
11	Deploying Windows 10 OS	●	
12	Creating snapshots	●	
13	Windows server 2016 IPV4 configuration	●	
14	Windows server 2016 roles and features	●	
15	Windows server users an groups	●	
16	Windows server active directory	●	
17	Windows server domain controller configuration	●	
18	Windows server groups and policies	●	
19	Windows server firewall configuration	●	
20	Windows server DNS configuration	●	

21	**Windows server DHCP configuration**	●	
22	**Completing Windows server hardening checklist**	●	
23	**Installing the bro network security monitor**	●	
24	**Installing shadow security scanner**	●	
25	**Installing Suricata**	●	
26	**Installing snort**	●	
27	**Join kali Linux to Windows server 2016 domain**	●	**Linux Machine keep disconnecting from the domain after shutdown.**
28	**Kali Linux users and groups**	●	
29	**Kali Linux SSH configuration**	●	
30	**Install Nmap**	●	
31	**Install Super-scan**	●	
32	**Install Hping3**	●	
33	**Install Nessus**	●	**The software is not stable, keeps crashing under heavy load**
34	**Install Xprobe2**	●	
35	**Install SARA**	●	
36	**Install Snort in windows 10**	●	
37	**Windows 10 firewall configuration**	●	
38	**Windows 10 user configuration**	●	
39	**Windows 10 network configuration**	●	
40	**Windows 10 joining windows server 2016 domain**	●	

Table 9 Installation status

Testing Chapter

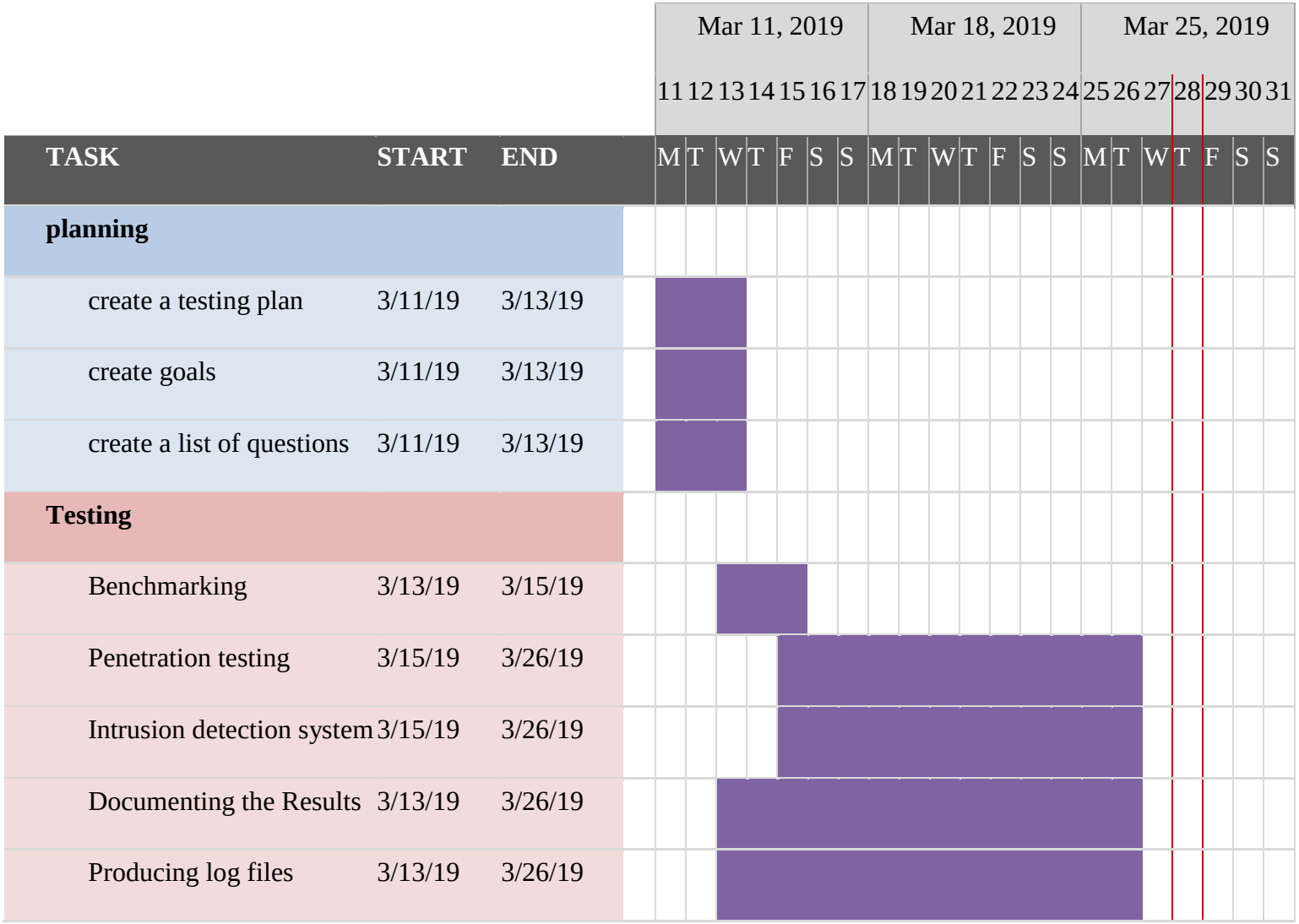
64 Overview

this is the testing of a virtualize environment, to study the possibility of using Intrusion detection systems and Penetration Testing software in Portable/Micro computing devices and then study the performance of these machines and compare them to the security appliances being used now days. Also, it is going to help in investigating the idea of the possibility of creating neural network with these devices and implementing them in a pre-existing network infrastructure.

65 Goal of the test

The goal of this experiment is to emulate a physical environment using VMware hypervisor that can handle Penetration testing scenarios and intrusion detection tests. Both the Banana Pi M64 VM and Raspberry Pi VM should be able to handle the penetration testing tools and should be able to keep up with the traffic when using intrusion detection software like Snort or BRO-IDS. The main goal is to show that a low-cost micro computing boards can be used as both an intrusion detection system and as a vulnerability assessment device at the same time effectively.

67 Testing Gantt chart



68 Design

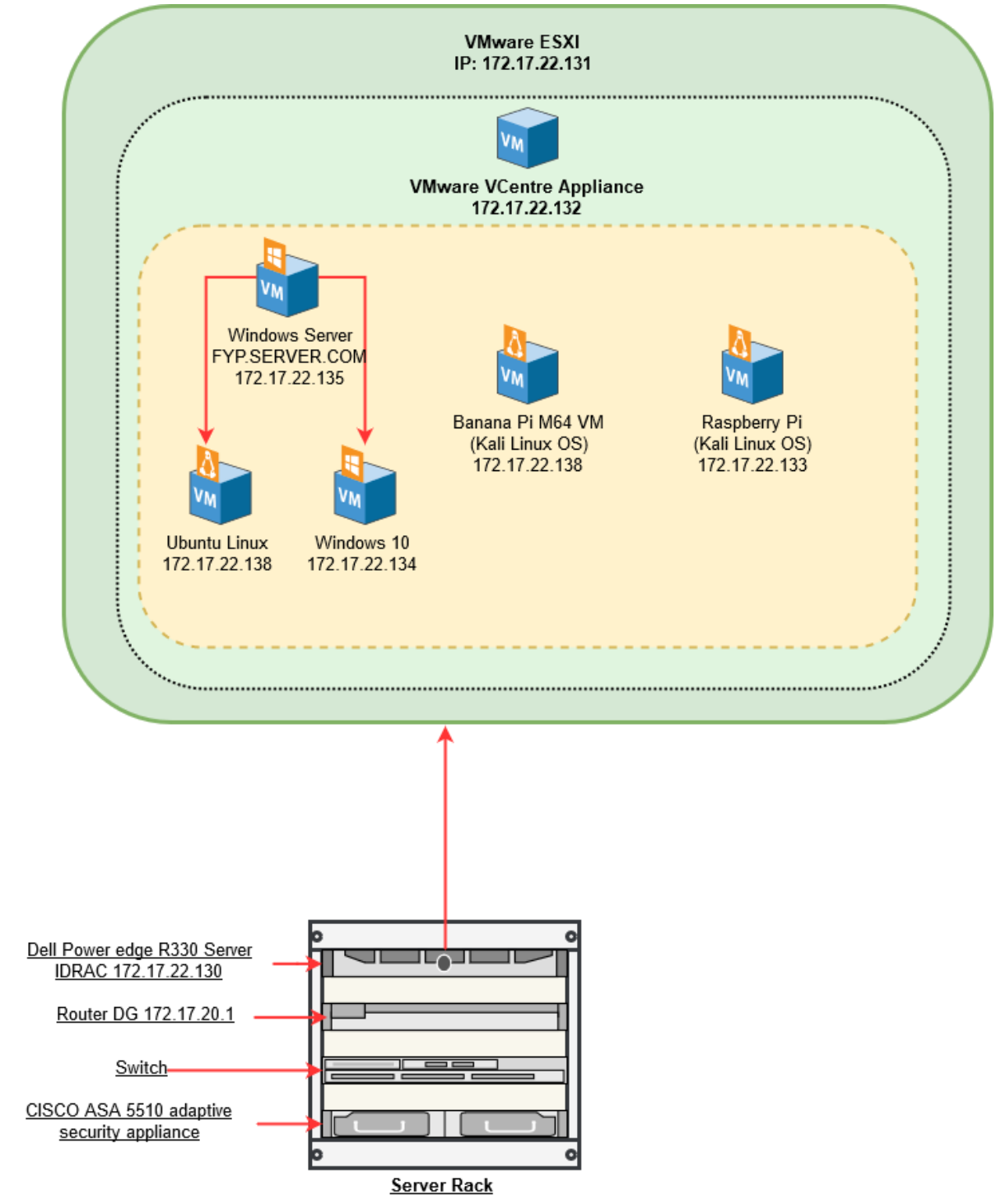


Figure 11 full design topology

69 Testing topology

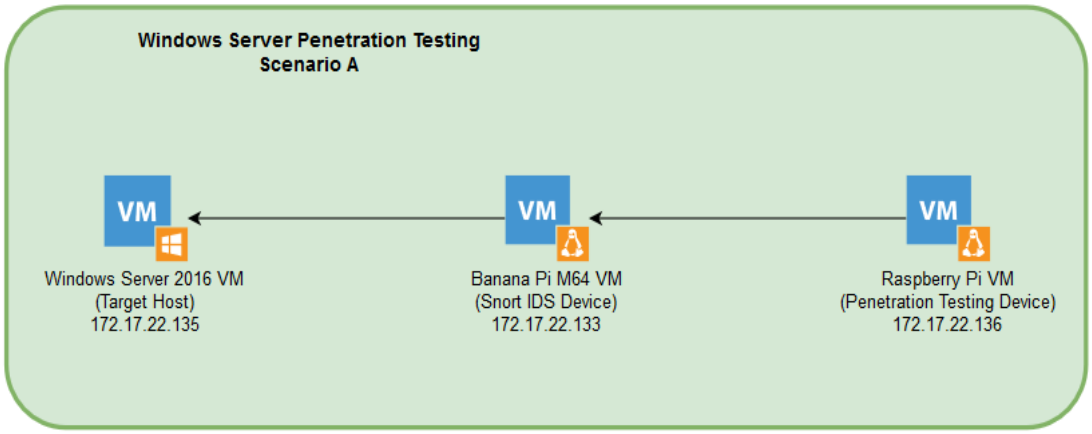


Figure 12 Windows VM Pen-Testing + SNORT IDS scenario A

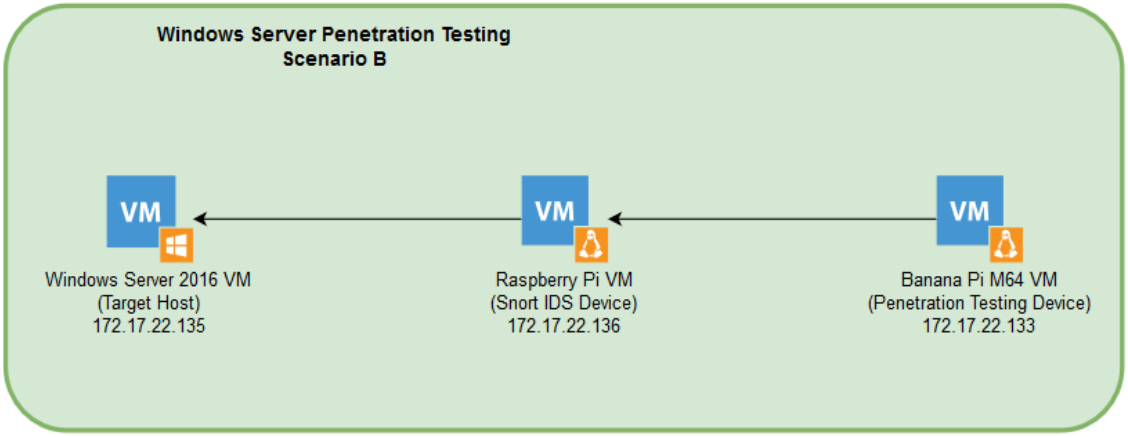


Figure 13 Windows VM Pen-Testing + SNORT IDS scenario B

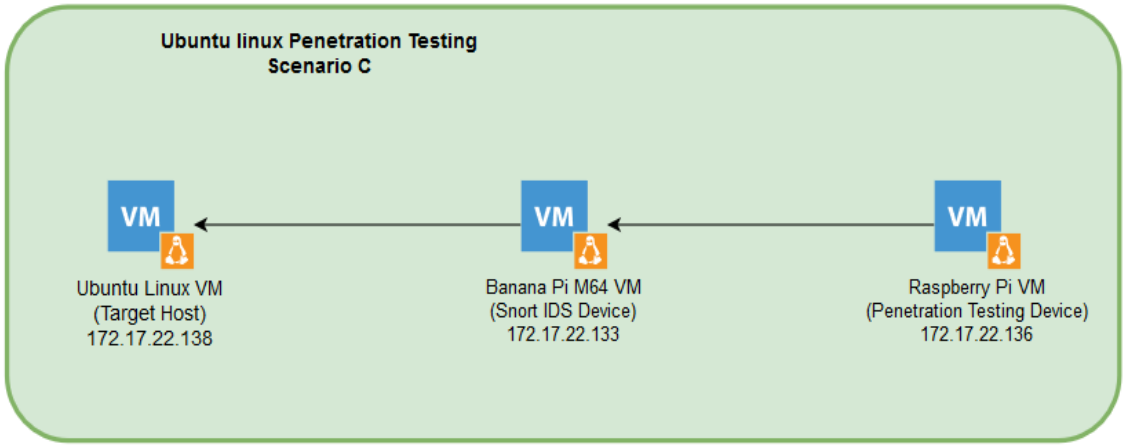


Figure 14 Ubuntu Linux VM Pen-Testing + SNORT IDS scenario C

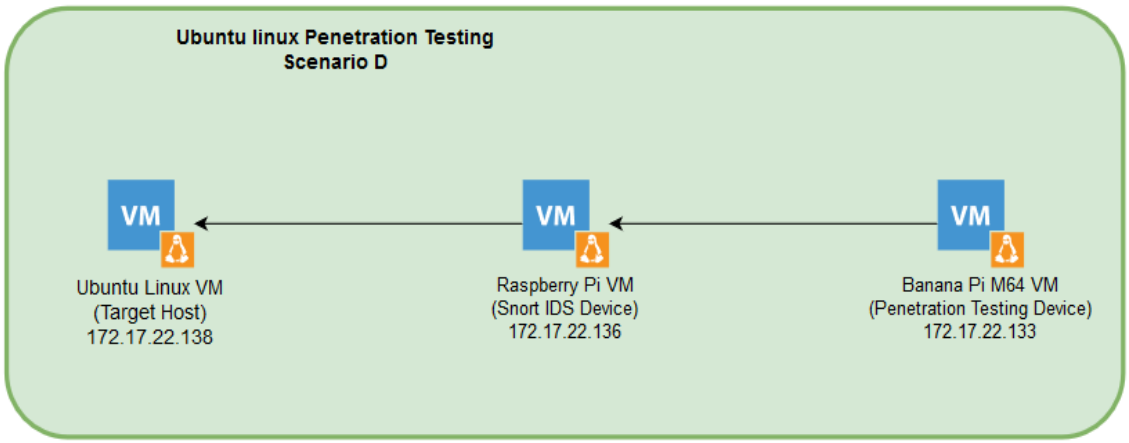


Figure 15 Ubuntu Linux VM Pen-Testing + SNORT IDS scenario D

70 Test methodology overview

The testing will be done in four different scenarios. Both the Raspberry Pi VM and Banana Pi M64 VM will take turn in each scenario to either be the Penetration testing device or the IDS device. Traffic will be routed from the Pen-testing device VM to the targeted host through the IDS Device VM using the native route command in Linux. Also, multiple tools will be used to log the activity of the VMs and log files will be produced to save the metrics of the system at that moment.

70.1 Windows VM Penetration testing & SNORT IDS

In scenario A, the Raspberry Pi VM will be the Pen-testing device and the Banana Pi M64 VM will be the IDS device with Snort installed in it. The targeted device will be a windows-based host. In scenario B, the Banana Pi M64 VM will be the Pen-testing device and the Raspberry Pi VM will be the IDS device with Snort installed in it. The target host will be a windows-based host.

During the testing both CPU and RAM usage will be monitored to produce a result that reflect the performance of the VMs during the test.

70.2 Ubuntu Linux VM penetration testing & Snort IDS

In scenario C, the Raspberry Pi VM will be the Pen-testing device and the Banana Pi M64 VM will be the IDS device with Snort installed in it. The targeted device will be an Ubuntu Linux VM. In scenario D, the Banana Pi M64 VM will be the Pen-testing device and the Raspberry Pi VM will be the IDS device with Snort installed in it. The target host will be an Ubuntu Linux VM. During the testing both CPU and RAM usage will be monitored to produce a result that reflect the performance of the VMs during the test.

71 Environment

All the simulation of the physical network was done in a private internal network within VMware ESXI virtualized environment. The Windows Server 2016 VM with 4GB of RAM and is fully configured with a firewall to imitate a real windows server instance, same system and firewall configuration was made to the Windows 10 VM that has 2GB of RAM and the Ubuntu VM that has a 2GB of RAM. The Raspberry Pi VM was configured with 1GB of RAM and 32GB of storage space and it is using Kali Linux as an operating system. The Banana Pi M64 VM was configured with 4GB of RAM 64GB of storage space and it is using Kali Linux as an operating system.

All the VMs share the same processor in the server which is Intel Xeon E3-1240 v5 Quad-core, and the server has a total of 16GB of RAM which is shared between all the VMs.

72 VMs specifications

Operating system	Device	RAM	Hard drive	CPU
Kali Linux	Raspberry Pi VM	1 GB	32 GB	Intel Xeon E3-1240 v5 Quad-core
Kali Linux	Banana Pi M64 VM	4 GB	64 GB	Intel Xeon E3-1240 v5 Quad-core
Windows server 2016	VM	4 GB	20 GB	Intel Xeon E3-1240 v5 Quad-core
Ubuntu	VM	2 GB	10 GB	Intel Xeon E3-1240 v5 Quad-core
Windows 10	VM	2 GB	20 GB	Intel Xeon E3-1240 v5 Quad-core

Figure 16 VM Specs

73 Purpose and questions

The main purpose of this study is to examine the effect on network performance when using a Raspberry Pi or Banana Pi M64 as an intrusion detection device and penetration testing device. The aim is to see how a Raspberry Pi VM compares to Banana Pi M64 VM and measure their capabilities to handle the network traffic while examining the traffic for intrusion and penetration testing. For the throughput to be within acceptable limits a requirement was made that there should be no loss greater than 30% - 40% and the CPU Usage should not go above 55%. This is intended as a point of reference used to evaluate and analyse the data gathered from the experiment. The tester will also measure the CPU and memory usage on the VMs so that the tester may be able to detect potential limitations in the hardware. The questions that's being asked are:

1. Can a Raspberry Pi or Banana Pi M64 be used as an intrusion detection system in a network without lowering the network performance outside of acceptable limits?
2. How would a Raspberry Pi VM compare to Banana Pi M64 VM in terms of throughput, CPU and memory performance?
3. Can a Raspberry Pi or Banana Pi M64 be used as a Penetration testing tools in a network without lowering the network performance outside of acceptable limits?

74 Goal of the testing

The goal of this experiment is to emulate a physical environment using VMware hypervisor that can handle Penetration testing scenarios and intrusion detection tests. Both the Banana Pi M64 VM and Raspberry Pi VM should be able to handle the penetration testing tools and should be able to keep up with the traffic when using intrusion detection software like Snort or BRO-IDS. The main goal is to show that a low-cost micro computing boards can be used as both an intrusion detection system and as a vulnerability assessment device at the same time effectively.

75 Virtual machines benchmarking

75.1 Kali Linux Benchmarking

Kali Linux benchmark is done using HardInfo tool, which allows me to quickly get an impression of the system performance which is important to test the system under intensive load.

The results of the benchmark are in the appendix (8.1)

75.2 Windows server 2016 benchmark

This benchmark was conducted using NetIO-GUI, Novabench and HardInfo. Netstat will be used to do stress test to measure the network speed, on the other hand HardInfo will allow me to quickly get an impression of the system performance which is important to test the system under intensive load

75.3 Windows 10 Benchmark

This benchmark was conducted using NetIO-GUI and HardInfo. Netstat will be used to do stress test to measure the network speed, on the other hand HardInfo will allow me to quickly get an impression of the system performance which is important to test the system under intensive load.

76 Testing

76.1 Penetration Testing

In this section multiple tools and methods have been used, both the Raspberry Pi VM and the Banana Pi M64 VM performed the Penetration testing without any issue. Bellow all the related aspect of the findings will be discussed.

76.1.1 Type of attacks

76.1.1.1 Nmap

The most popular open source tool used to scan hosts and services on a network is Nmap. Nmap's advanced features can detect different applications running on systems as well as offer services such as the OS fingerprinting features. Nmap can be very effective; however, it can also be easily detected unless used properly.

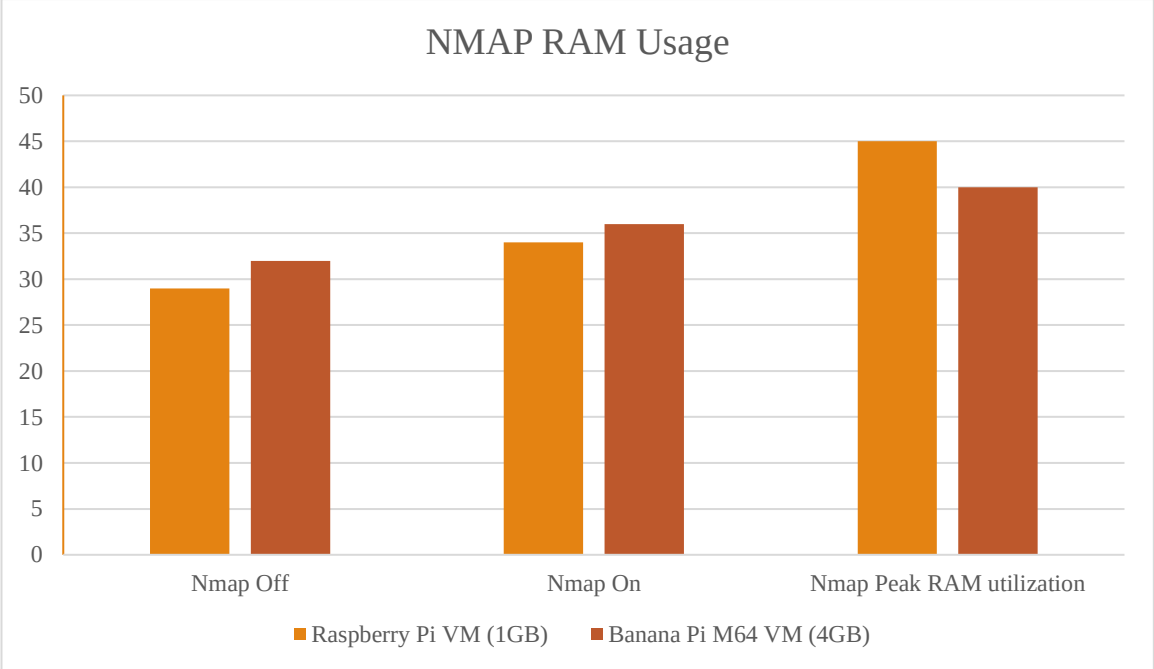


Figure 17 Nmap RAM utilization

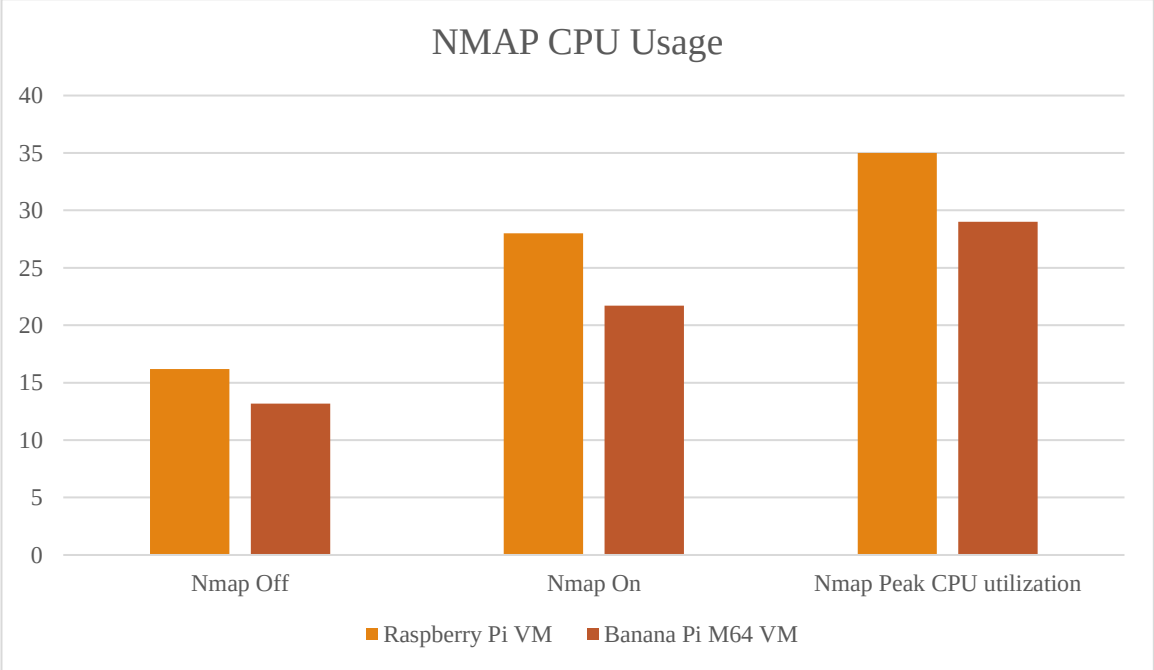


Figure 18 Nmap CPU utilization

Nmap was used in both the Raspberry Pi VM and the Banana Pi M64 VM, the CPU utilization in the raspberry pi VM was at 28.2% and it peaked at 35% which didn't affect the performance of the VM. On the other hand, the Banana Pi VM ran very good since the CPU Utilization was at 21.7% and peaked at 29% which still under 50% utilization which means the performance of the VM won't be affected by having Nmap running for

the penetration testing. The run time was 15 minutes with raspberry pi VM and 13 minutes with Banana Pi M64 VM.

When Nmap as used, it ran against the Windows Server 2016 VM using the IP address 172.17.22.135 and the Ubuntu VM using the IP address 172.17.22.138, both VMs firewall was fully configured and all the ports were closed except the service ports that are open for the operating system services. All the result logs and the screenshots from the scan are included in the appendix.

76.1.1.2 Sparta

SPARTA is a python GUI application that simplifies network infrastructure penetration testing by aiding the penetration tester in the scanning and enumeration phase. It allows the tester to save time by having point-and-click access to their toolkit and by displaying all tool output in a convenient way.

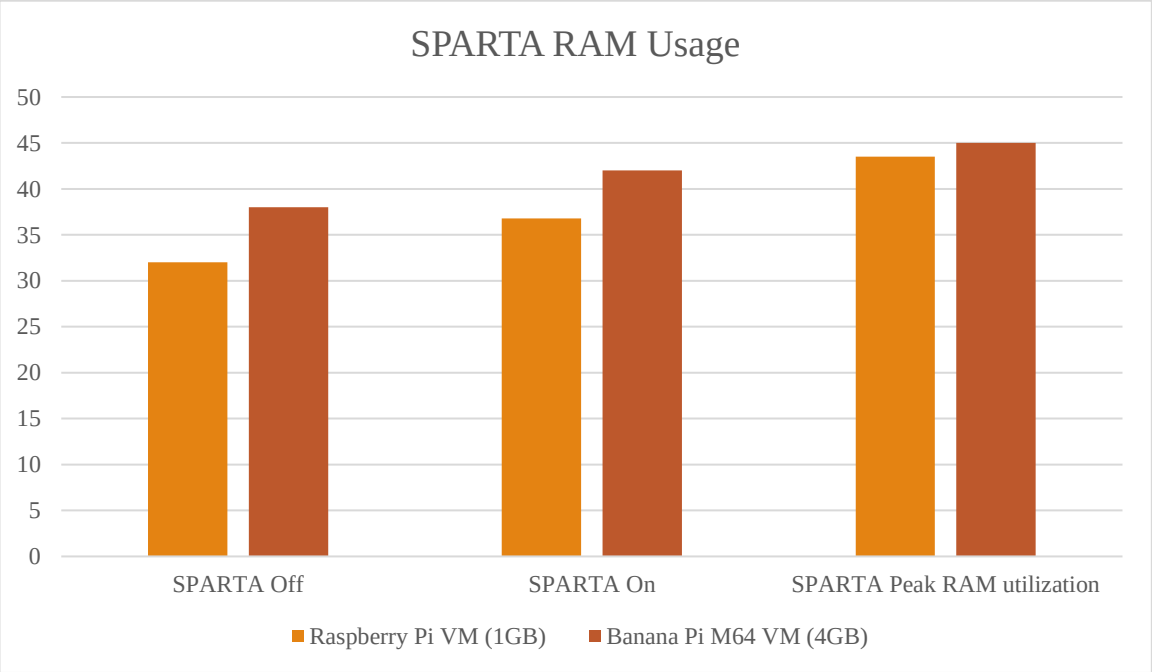


Figure 19 SPARTA RAM utilization

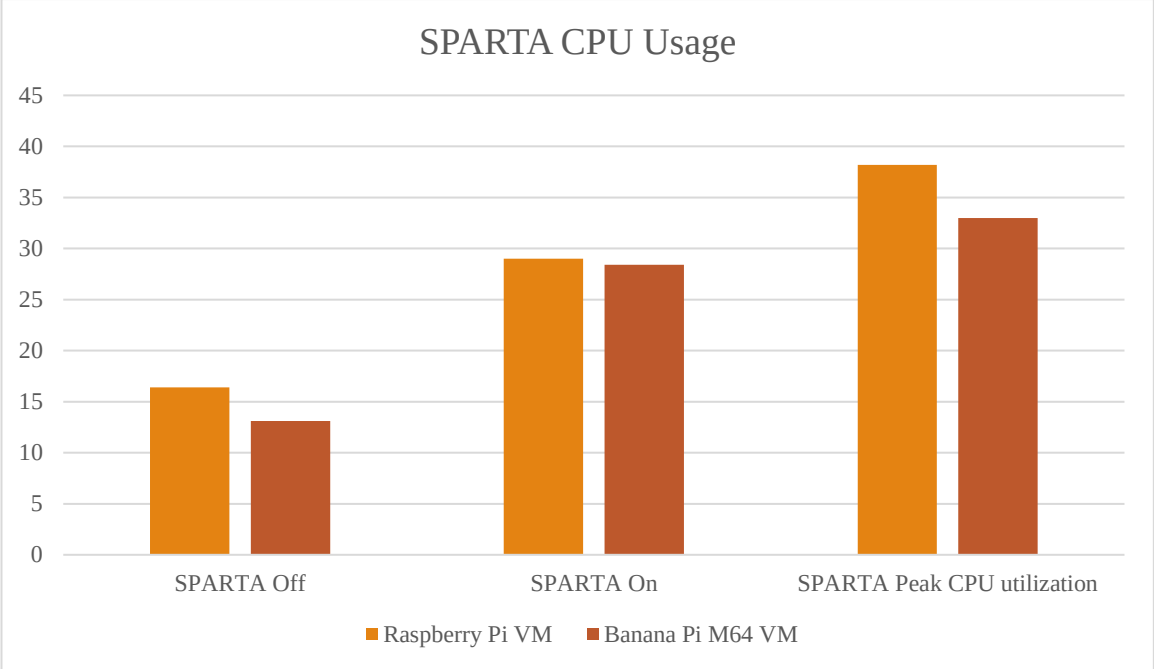


Figure 20 SPARTA CPU utilization

From the above chart it is noticeable that SPARTA require more processing power compared to Nmap, and the reason could be that SPARTA uses Python scripts to run the scans and it also uses a GUI rather than a command line like Nmap. Although SPARTA uses more CPU and Memory resources having SPARTA running didn't affect the performance of the VMs, it took more time for the scan to be done compared to Nmap. SPARTA took approximately 18 minutes to do a full scan in the Raspberry Pi VM and Banana Pi M64 VM 17 minutes to finish the scan. The run time was 18 minutes with raspberry pi VM and 17 minutes with Banana Pi M64 VM.

SPARTA was used to scan all the TCP and UDP ports, also it was used it to scan the Operating system of both the targeted VMs. All the results of the open ports and all the information related to the scan are in the appendix.

76.1.1.3 Armitage Kali

Armitage Kali is a tool used for scanning a targeted host and find any exploitation that can be used against that host. Armitage tool ran this against the Windows Server 2016 VM using the IP address 172.17.22.135 and the Ubuntu VM using the IP address 172.17.22.138. The tool uses its large database of known exploitations against the hosts and it uses multiple methods of scanning to create or find a weak spot in the targeted host that can be used by me to exploit the host.

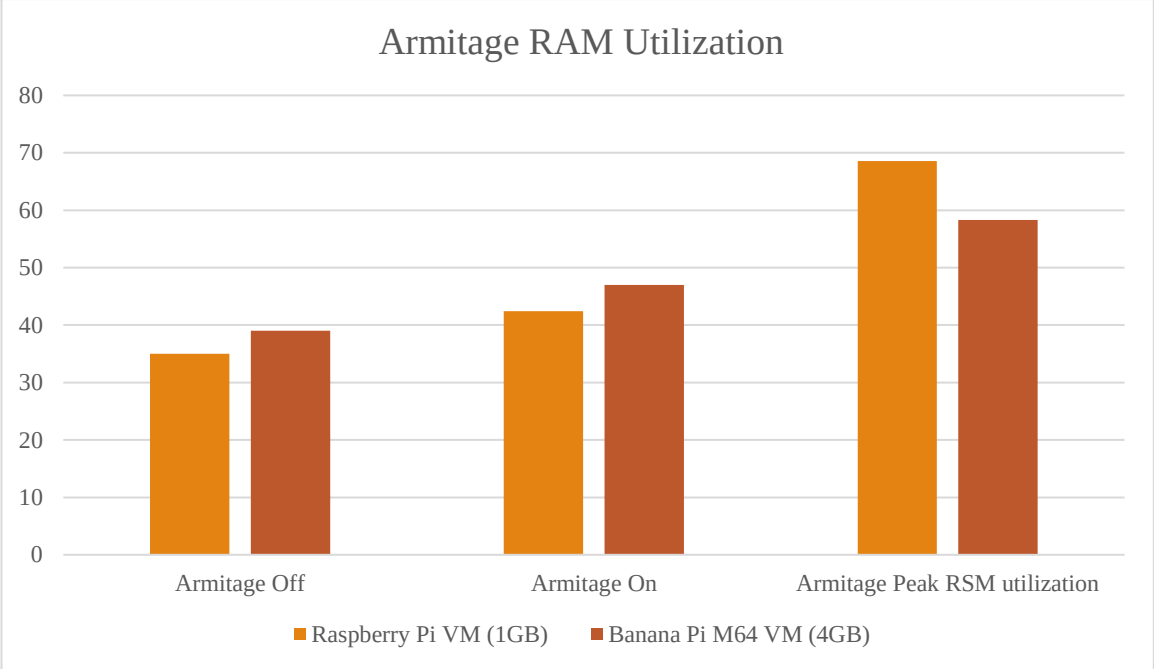


Figure 21 Armitage RAM Utilization

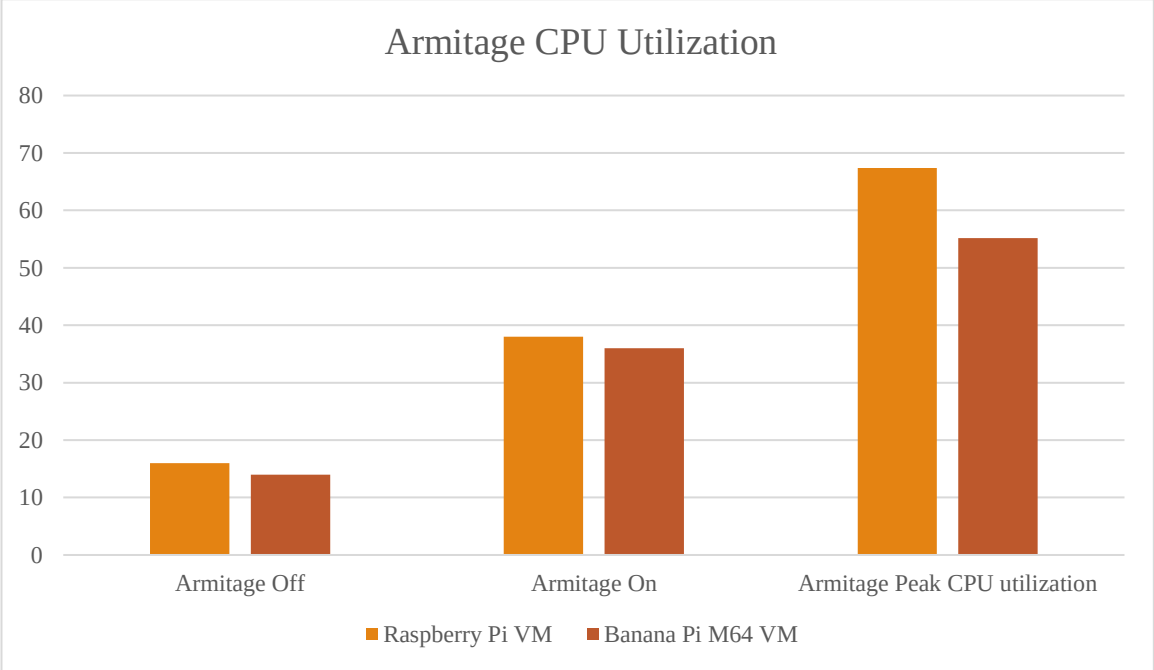


Figure 22 Armitage CPU Utilization

The full scan took 4:24:657 to finish in the Banana Pi M64 VM, and 5:13:784 to finish in the Raspberry Pi VM. From the chart above Armitage CPU utilization is very high compared to the two previous tools, in bot VMs the CPU Utilization went above 60% which made both VMs very slow to use when Armitage scan is running.

The results of the scans were impressive, all the open ports were detected and all the information about the target host, after the scan is done Armitage will find all types of exploitations that can be found and then used to attack the host. Unfortunately, Armitage found few exploits that can be used, but using them to attack the host was beyond the knowledge of the tester and couldn't go further after finding the exploits.

The log files and screenshots produced by Armitage are in the appendix.

76.1.2 Frequency and quantity of the attack

The frequency of the attacks was altered to ensure that the processing limitations of the Raspberry Pi VM and Banana Pi M64 VM could be measured and applied to a real-world scenario. By increasing the frequency of attacks, the IDS should be able to determine that the attacks are more frequent and not regard the packets as an increase in normal traffic.

76.2 Intrusion Detection Software (Linux\Windows)

The approach for this phase is inductive and data will be collected from the tests. Data will be gathered by performing several tests and it will be used as the foundation for this project. The data that will be measured is the CPU usage, memory usage and bandwidth throughput and the effectiveness of the IDS software's. An observation of the results will be made, and from that an overall conclusion will be discussed below.

76.2.1 BRO IDS

BRO-IDS ran into multiple issues, first or all the software used all the memory available which led the VMs to crash even when the RAM was increased to 4GB the VMs still crashed. From what was provided by BRO-IDS website, the BRO-IDS needed minimum of 4GB which in this case running the BRO-IDS left the VMs with no memory to use. Unfortunately, due to this issue this software couldn't be used for intrusion detection.

76.2.2 SNORT

Snort is a network IDS. As Snort is an open source software it is supported on several operating systems. Snort a versatile software as it is capable of analysing traffic in real-time and can be used to perform several different functions such as traffic analysing and packet sniffing. These functions can be used to detect and prevent attacks against the network (snort, 2019). Snort applies signatures and rules to find malicious network traffic and, in this testing, two sets of rules one with full list of rules enabled and the other one with ten rules enabled, both sets of rules can be viewed in the appendix.

Invoking snort alert will be done using the Nmap, Armitage, Hping, Sparta and some other command line tools. Traffic generated by those software's would include SAMBA,3HTTP, UDP and AppleTalk also the data sets generated by these tools also include some captures of a teardrop attack, DNS exploits, worms including the slammer worm, and the DNS remote shell anomaly. Once sent Snort should be able to detect them and create an alert and log the alert for the network administrators. Regular application traffic was also submitted to the network to show that the IDS would not confuse it for malicious packets.

Both systems were attacked at the same time by the different tools mentioned above. When an alert was logged, the system would increase the alert count. ICMP and UDP heartbeat negatives were sometimes classified as alerts in the SNORT log. The SNORT dump from Appendix A shows that all packets on the scans in the Raspberry Pi VM machine were caught and processed same thing happened using the Banana Pi M64 VM. In Appendix B, the total percentage of scanned packets was 100%. No packets were dropped or let through without being scanned. All packets were accounted for in the SNORT logs, and both the VMs were able to keep up with the traffic volume.

76.2.3 Network Throughput

This section will show the results of the throughput test between the Raspberry Pi VM and Banana Pi VM while having snort enabled.

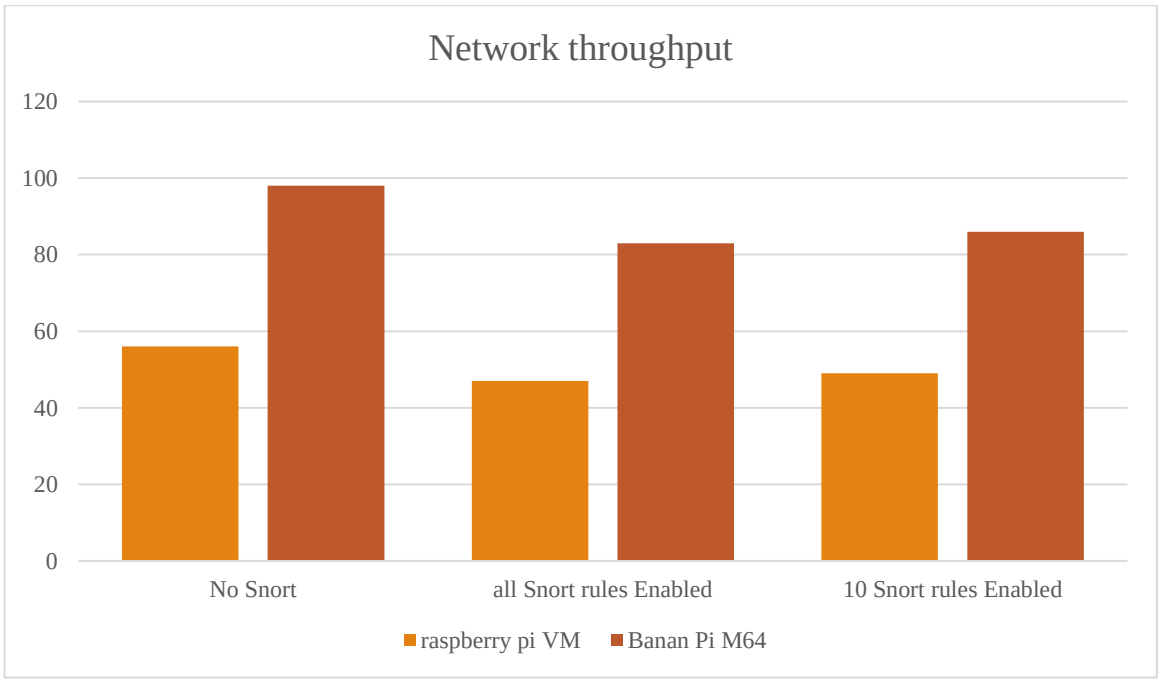


Figure 23 Network throughput

The graph in figure 4 shows the average network throughput through each of the Raspberry Pi VM and Banana Pi M64 during different set of conditions. The Y-axis shows the throughput measured in Mbit/s and the X-axis displays how many Snort-rules was implemented as a set of conditions. With all rules implemented, the throughput of Raspberry Pi VM was measured to 48.86 Mbit/s and Banana Pi M64 to 83.58 Mbit/s. With ten rules implemented in snort, Raspberry Pi VM measured to 49.1 Mbit/s and Banana Pi M64 VM to 86,8Mbit/s. With Snort offline the throughput was measured to 56.7 Mbit/s for the Raspberry Pi VM and 94.02 Mbit/s for Banana Pi M64 VM.

The results from the baseline throughput tests shows that there is a difference in throughput between the Raspberry Pi VM and the Banana Pi M64 VM. The tests also show that there is a throughput degradation when activating Snort, but once activated the number of Snort rules did little to change the network performance for any of the VMs. It is unknown to me if Snort could be optimized to process traffic faster and therefore grant higher throughput, even after few researches no evidence found to indicate the ability to optimize Snort Network traffic.

76.2.4 CPU Load

The CPU load will be unreliable as it is a virtualized environment, so all VMs share the same processor making it unreliable and it was proven to be inconsistent.

76.2.5 Memory load

This section will show the results of the memory performance measurements while testing the throughput capacity on Raspberry Pi VM with 1GB RAM and Banana Pi M64 4GB RAM.

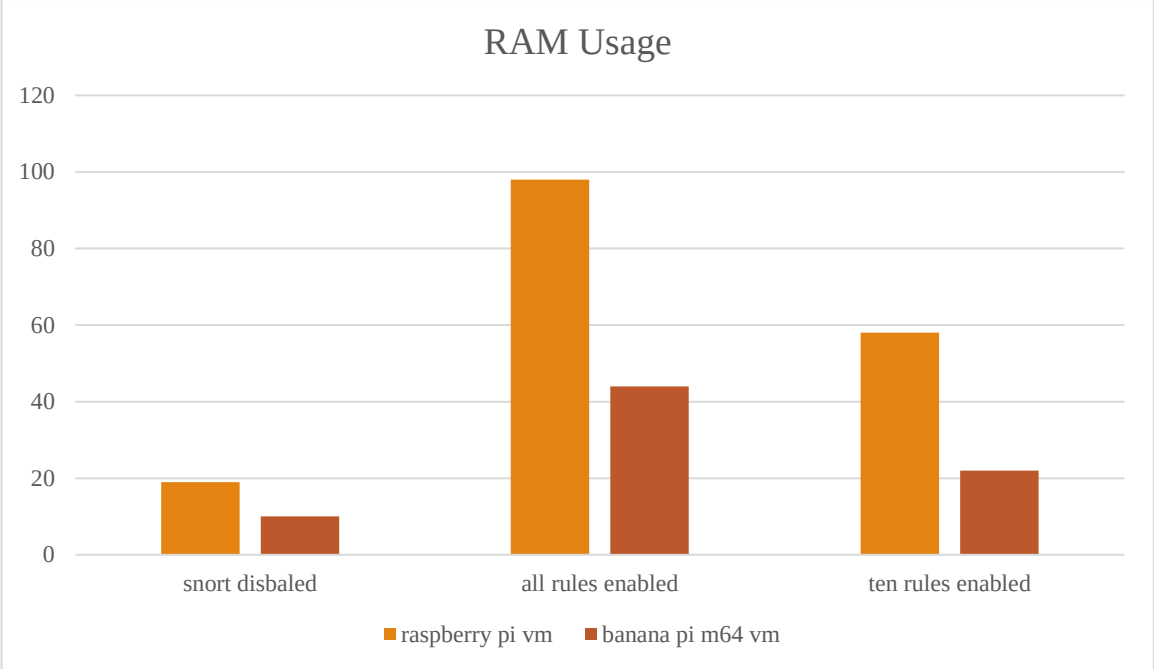


Figure 24 RAM Usage

The graph in figure 5 displays the average memory load of each of the Raspberry Pi VM and Banana Pi M64 VM during the testing of the throughput, under three different sets of conditions. The Y-axis shows the memory load measured in percent and the X-axis shows the set of conditions with how many rules are implemented. With all rules implemented the memory load on Raspberry Pi VM was measured to 98.34% and 44.08% on the Banana Pi M64 VM. With ten rules implemented the memory load was measured to 58,416%on the Raspberry Pi VM and 22.268% on the Banana Pi M64 VM. With no rules implemented and Snort offline the memory load on the Raspberry Pi VM was19.5% and 10,172% on the Banana Pi M64 VM.

76.3 IDS Conclusion

This project examined the possibilities of using a Micro Computing unit as an IDS in an IoT network by using a virtualize environment. The premise was to measure the performance of Virtual machine mimicking the Portable devices and see if they could satisfy the requirements of not reducing network through-put by more than 30% - 40%.

Using the minimum specs virtual machine mimicking a raspberry pi as an IDS did have a negative effect on the network throughput. All though the effect was less so on a VM mimicking Banana Pi M64, because of its improved hardware over a raspberry pi it is evident that the extra RAM and CPU power play a big role in increasing the performance. Even with the IDS software disabled the best throughput performance measured in the tests was 94 Mbit/s with the Banana Pi M64 VM. It could be argued that on a 100 Mbit/s connection a loss of only 6 Mbit/s could be acceptable foremost users. The Raspberry Pi VM suffered a throughput drop which exceeds the limit of 30% regarding throughput performance, but it could still be used as an IDS if it fulfils the user's requirements. One scenario could be the following: the IDS is connected between the Internet and the user's network. If the user has an Internet connection of 14Mbit/s, there would be no need for any greater throughput capacity of the IDS device. The Raspberry Pi would then be a better choice because of the lower implementation costs compared to the Banana Pi M64.

In 2013 (Kho, 2019) a study was made on the possibility on using a Raspberry as an IDS in a home network. They performance tested a Raspberry Pi model B+ running the operating system IP Fire and the intrusion detection software Snort. The results from this study showed that a Raspberry Pi could be used as an IDS, but with the following problems:

1. A limit on how many Snort-rules could be used due to limitations in memory.
2. A noticeable degradation in throughput when Snort was active.

When comparing this project with the previous study from 2013 it was possible in this scenario to avoid the limitations they mentioned, while getting throughput results that fulfil this project requirements. However, while they had a requirement that the implementation should be as easy as possible which they had to take consideration when choosing operating system. However, on their study it is thought that they over complicated the selection of the OS and the requirements needed for this to be implemented in a network perhaps because the study was done in 2013 and they lacked the improved operating systems we have now days as their biggest issue was resource

management by the OS but this does not have to be a downside nowadays, by removing the graphical user interface from any OS it will reduce the amount of resources needed to operate it so instead a potential is seen in the market for companies wishing to sell this kind of implementation as a low-cost service and a big potential for increased security in smart infrastructures and IoT enabled environments. For example, the company could prepare a memory card with all the necessary configurations and updates. The user would then only need to connect the device to the network, insert the memory card and power adapter. This solution could even be used by larger companies wishing to improve the network security of users working from their homes.

76.4 Penetration Testing Conclusion

This project examined the possibilities of using a Micro Computing device as penetration testing in an IoT network by using a virtualize environment. The premise was to measure the performance of Virtual machine mimicking the Portable devices and see if they could be used as penetration testing devices and IDS devices at the same time.

Both devices were able to complete all the vulnerability penetration tests without a single failure. The Raspberry Pi VM suffered a high CPU utilization which peaked over 60%, but it is believable that it can still be used as a pen testing device. On the other hand, The Banana Pi VM didn't suffer from any lag or high CPU utilization because the Banana PI VM has 4GB of RAM compared to 1GB of RAM in the Raspberry Pi VM. Unfortunately, the high CPU utilization will affect the IDS process and any other process that needs to run at the same time. This is a big disadvantage for the Raspberry Pi VM, and it highlight a high risk that the Raspberry PI can shut down due to high CPU utilization at any time. Moreover, it was noticed that the run time for each pen-test differed between the VMs. Generally, the Raspberry Pi VM took longer time to complete every single test compared to the Banana Pi M6 VM. Processing the test at a faster rate means the resources will be available faster for other test runs and it won't affect the IDS process for a long time. The shorter the run time for any test the better it is, for example due to the fast processing time in the Banana pi, more tests can be conducted in one hour compared to what a Raspberry Pi can do in the same time.

At the end, it is advisable to use the Banana Pi M64 for pen-testing as it can handle more processing than the Raspberry Pi.

77 Results Summary Table

Test type	Status	notes
Benchmark Windows server 2016 VM	●	
Benchmark Kali Linux VM (Raspberry PI)	●	
Benchmark Kali Linux VM (banana PI)	●	
Benchmark windows 10	●	
Record network throughput	●	
Record RAM usage	●	
Use Nmap for network scanning	●	
Use SPARTA for scanning	●	
Use Armitage	●	The tool is working fine, but to continue and use the exploits is out of the project scope
Check if SNORT is working	●	
Use BRO-IDS	●	The VM is unable to handle the software as it need too much resources
Record CPU utilization	●	

Table 10 Results Summary Table

Case study chapter

78 Overview

This chapter will cover multiple aspect of implementing this system in a real-life scenario. A comparison will be made between Raspberry PI, Banana Pi M64 and a regular IDS system when it comes to cost and power usage.

79 Cost comparison

Cost of purchase is a big motive for the buyer, the higher the cost the harder it is to sell such a product. In here a comparison is made between Raspberry Pi, Banana Pi M64 and a general IDS system when it comes to buying 15 network IDS and 15 Host IDS for medium to large enterprise.

	Raspberry Pi	**Banana Pi M64**	**General IDS**
Device price	€35 - €45	€75-€85.36	€10,000 per network €1000 per host €5000 management station
Storage 32GB	Samsung 32GB Evo €15.99	Samsung 32GB Evo €15.99	-
Storage 64GB	SanDisk SDXC Ultra €21.90	SanDisk SDXC Ultra €21.90	-
Storage 128GB	Not supported	Samsung 128GB Evo Plus €50.99	-
Power cables	RP Power Supply €8.99	Power Supply €16.55	-
Cost of 15 hosts IDS	€988.35 (assuming 64GB storage, €35 device price)	€1,776.75 (assuming 64GB storage, €80 device price)	€,1500
cost of 15 Network IDS	€988.35 (assuming 64GB storage, €35 device price)	€1,776.75 (assuming 64GB storage, €80 device price)	€150,000
Total price	€1976.7	€3553.5	€170,000

Table 11 Single board devices cost

80 Cost of designing, implementation, staff and maintenance

	Raspberry Pi	Banana Pi M64	General IDS
Management cost	€5000-€7000	€7000-€10,000	€225,000-€300,000
Design cost	€10,000-€20,000	€10,000-€20,000	€15,000-€80,000
Maintenance	15%-25% of the total cost of purchase	15%-25% of the total cost of purchase	15%-25% of the total cost of purchase
Engineer Cost	€75,000 (€60,000 salary plus €15,000 benefits & admin) x1-2	€75,000 (€60,000 salary plus €15,000 benefits & admin) x1-2	€75,000 (€60,000 salary plus €15,000 benefits & admin) x2-5
Group Manager Cost	€100,000 (€80,000 salary plus €20,000 benefits & admin) x1	€100,000 (€80,000 salary plus €20,000 benefits & admin) x1	€100,000 (€80,000 salary plus €20,000 benefits & admin) x1

Table 12 Cost of designing, implementation, staff and maintenance

81 Power consumption

Power consumption is a big factor nowadays when it comes to deciding which device is more appropriate for the environment since majority of the enterprises aim to get a green environment friendly badge. In here an assumption is made that the devices ran for one year accumulating 8760 hours of up time with no down time assuming that the devices were idle for 15% of the time giving us a total of 1314 hours of ide time, another assumption is made that the deice ran on typical usage for 50% of the time giving us 4380 hours of typical use and 35% of maximum usage with a total of 3066 hours. According to price provided by money guide Ireland the price for 1Kwh is €.0.2486 cents.

	Raspberry pi	Banana pi m64
Idle load	2 watts/h	3 watts/h
Typical load	2.3 watts/h	3 watts/h
Maximum load	3 watts/h	4 watts/h
Total cost	€5444.34 per year	€7295.4156 per year

Table 13 power consumption

Discussion

From the testing results it was noticeable that there is a difference between the raspberry pi and the Banana Pi M64 when it comes to network throughput, CPU utilization and RAM utilization. Generally, the Banana pi m64 performed better than the Raspberry pi in all the tests that were conducted. When comparing the difference between the two devices, it is noticeable that there is a small difference in numbers between the two devices. It can be argued that the performance of these devices can increase dramatically by removing the GUI from the operating system. By removing the GUI, the amount of resources used by the OS will decease massively and such increase in the availability of the resources will increase the performance and decrease the load on the device.

Moreover, since these devices will be performing as an IDS majority of the time the processing power is a big concern. It was observed in a rare occasion during testing that when the VMs go under heavy load and there is a high CPU and RAM Utilization the network throughput starts to throttle, and network packets start to drop. Such thing was happening more in the raspberry pi VM compared to the banana pi m64 VM because the Banana pi VM has 4GB of RAM compared to the raspberry pi VM which has 1GB of RAM.

Furthermore, there are two limitations with the raspberry pi and banana pi devices:

1. The limited amount of disk space.
2. The longevity of the SD cards.

Currently raspberry pi only supports SD cards up to 32GB and 64GB as a micro SD compared to the banana pi m64 it can handle 32GB, 64GB, 128GB SD cards. SD cards form the basis of the stable storage of the raspberry pi and banana pi m64. A typical IDS software can have multiple log files running at once, and it can read and write constantly on each log file. The number of read and writes increases significantly with the increase of network packet monitoring which might lead the SD card to fail or even decrease the life time of the SD card. Such issue can be fixed by writing the log files to a remote server or a physical logging device that could be located physically with the raspberry pi or the banana pi m64 and connected via USB but using a network log storage that can be easily configured using Docker containers or using Linux rsyslog. This would both save space on the disk and reduce the amount of reads and writes to the SD card. However, using the

latter method of using any remote logging services would increase the amount of network traffic and could increase the load on the devices.

Conclusion

This thesis has shown that a low cost, low power device can be used as a viable alternative to traditional Intrusion Detection Systems. The cost of the system is substantially lower than a traditional IDS and the Raspberry pi or the banana pi m64 can be adapted due to the low cost and low power consumption. Moreover, this project examined the possibilities of using a single board Computing unit as pen-testing device and as an IDS in an IoT network by using a virtualize environment and the results helped in answering the three main goals of this project.

Firstly, it was proven from the results that both devices can operate as an intrusion detection system. Using the raspberry pi as an IDS did have a negative effect on the network throughput to some degree. All though the effect was less so on a VM mimicking Banana Pi M64, because of its improved hardware over a raspberry pi it is evident that the extra RAM and CPU power play a big role in increasing the performance. The raspberry pi performance can increase, and the network throttling can be somewhat fixed by removing the GUI from the operating system and removing all the unnecessary tools and application which come preinstalled in the OS. Although the Raspberry Pi VM suffered a throughput drop which exceeds the limit of 30% regarding throughput performance, it could still be used as an IDS if it fulfils the user's requirements specially in environments where latency in not important. So, it can be argued that The Raspberry Pi would then be a better choice in some occasions because of the lower implementation costs compared to the Banana Pi M64 if latency in not an issue to the network environment.

Secondly, this project examined the possibilities of using a Micro Computing device as penetration testing in an IoT network by using a virtualize environment. Both devices were able to complete all the vulnerability penetration tests without a single failure. The Raspberry Pi VM suffered a high CPU utilization which peaked over 60%, but it still can be used as a pen testing device. On the other hand, The Banana Pi VM didn't suffer from any lag or high CPU utilization because the Banana PI VM improved RAM of 4GB compared to raspberry pi which has 1GB of RAM. Generally, the Raspberry Pi VM took longer time to complete every single test compared to the Banana Pi M6 VM. Processing

the test at a faster rate means the resources will be available faster for other test runs and it won't affect the IDS process for a long time. Furthermore, the low RAM in raspberry pi is a significant disadvantage because it means the system will not be able to analyse data packets as fast as possible which will cause the network to throttle and it will become a bottleneck in the network environment.

Lastly, both devices lack the power to operate a full IDS software like BRO-IDS due the devices specs. The lack of RAM is big disadvantage in both devices, having 4GB of RAM is not enough nowadays since majority of the IDS software require a complete minimum of unused 4GB of RAM. This disadvantage makes it impossible to install any powerful Intrusion detection software in these devices, but Snort was more than capable to handle all the network traffic without dropping any packets. Although it is impossible to install powerful Intrusion detection software in raspberry pi or banana pi m64, it is possible to install Snort and combine it with other tools which will give extremely identical results to the other Intrusion detection software. Furthermore, using more high-end single board computing devices will make it possible to use high end IDS solutions which will cost more to implement but it will give a similar performance and efficiency to the IDS used nowadays in medium to large enterprises.

At the end, it is concluded that the price is one of the strongest arguments for this kind of implementation. The low cost of the Raspberry Pi or the Banana Pi means that it can be a more affordable IDS and penetration testing system compared to what would otherwise be an expensive enterprise grade solution. This solution opens several possibilities for users in increasing their network security from intrusions using low priced and affordable micro computing devices. Finally, it is advisable to use the Banana Pi M64 for such implementation as it can handle more processing than the Raspberry Pi.

82 References

Botta, A., de Donato, W., Persico, V. & Pescapé, A., 2016. Integration of Cloud computing and Internet of Things. *Future Generation Computer Systems,* march, 56(C), pp. 684-700 .

Whitmore, A., Agarwal, A. & Xu, L., 2015 . The Internet of Things--A survey of topics and trends. *Information Systems Frontiers,* 2 april, 17(2), pp. 261-274.

A. A.-F.et al., 2015. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys & Tutorials,* 15 june, 17(4), pp. 2347 - 2376.

Abduvaliyev, A. et al., 2013. On the Vital Areas of Intrusion Detection Systems in Wireless Sensor Networks. *IEEE Communications Surveys & Tutorials,* 15(3), pp. 1223 - 1237.

Altaf, I., Rashid, F. u., Dar, J. A. & Rafiq, M., 2015. *Vulnerability Assessment and Patching Management.* Faridabad, India, IEEE.

Anantvalee, T. & Wu, J., 2007. *A Survey on Intrusion Detection in Mobile Ad Hoc Networks.* Boston, MA: Springer.

Ashton, K., 1998. *From Internet of Data to Internet of Things.* s.l., s.n., p. 2.

Asplund, M. & Nadjm-Tehrani, S., 2016. Attitudes and Perceptions of IoT Security in Critical Societal Services. *IEEE Access,* Volume 4, pp. 2130 - 2138.

atzori, l., lera, a. & morabito, g., 2010. The Internet of Things: A survey. *Computer Networks.*

Chen, C.-K., Zhang, Z.-K., Lee, S.-H. & Shieh, S., 2018. *Penetration Testing in the IoT Age,* s.l.: IEEE.

Chen, L. et al., 2017. Robustness, Security and Privacy in Location-Based Services for Future IoT: A Survey. *IEEE Access,* Volume 5, pp. 8956 - 8977.

Chunlei, W., Li, L. & Qiang, L., 2014. *Automatic fuzz testing of web service vulnerability.* Nanjing, China, China, IET.

Commission, I. E., n.d. *Internet of Things: Wireless Sensor Networks,* s.l.: International Electrotechnical Commission.

CoreRFID, 2017. *THE INTERNET OF THINGS:PRACTICAL THOUGHTS FOR BUSINESS,* s.l.: CoreRFID.

Dao, N.-N.et al., 2017. Achievable Multi-Security Levels for Lightweight IoT-Enabled Devices in. *IEEE Access,* Volume 5, pp. 26743 - 26753.

Fremantle, P., 2010. *A Reference Architecture For The Internet of Things,* s.l.: WSO2.

Gartner, 2016. *Gartner Says By 2020, More Than Half of Major New Business Processes and Systems Will Incorporate Some Element of the Internet of Things,* s.l.: Gartner.

Giuliano, R., Mazzenga, F., Neri, A. & Vegni, A. M., 2017. Security Access Protocols in IoT Capillary Networks. *IEEE Internet of Things Journal,* 4(3), pp. 645 - 657 .

Hadjichristofi, G., Andrea, I. & Chrysostomou, C., 2015. Internet of Things: Security Vulnerabilities and challenges. *The 3rd IEEE ISCC 2015 International Workshop on Smart City and Ubiquitous Computing Applications.*

Hezam, A. A.-g., Konstantas, D. & Mahyoub, M., 2018. A Comprehensive IoT Attacks Survey based on a Building-blocked Reference Mode. *International Journal of Advanced Computer Science and Applications,* 9(3).

Hinai, S. A. & Singh, A. V., 2017. *Internet of things: Architecture, security challenges and solutions.* s.l., IEEE.

Hossain, M. S. et al., 2016. Toward end-to-end biomet rics-based security for IoT infrastructure. *IEEE Wireless Communications,* 23(5), pp. 44 - 51.

Hussain, A., FarooqiFarrukh & Khan, A., 2004. Intrusion Detection Systems for Wireless Sensor Networks: A Survey. *International Conference on Future Generation Communication and Networking,* pp. 234-241.

Hwang, Y. H., 2015. IoT Security & Privacy: Threats and Challenges. *IoTPTS '15 Proceedings of the 1st ACM Workshop on IoT Privacy, Trust, and Security.*

ITU, T. S., 2012. *SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-GENERATION NETWORKS.* s.l.:s.n.

Kaps, J.-P., 2006. *Cryptography for Ultra-Low Power Devices,* WORCESTER: WORCESTER POLYTECHNIC INSTITUTE.

Khan, R., Khan, S. U., Zaheer, R. & Khan, S., 2012. *FUTURE INTERNET: THE INTERNET OF THINGS ARCHITECTURE, POSSIBLE APPLICATIONS AND KEY CHALLENGES.* Islamabad, Institute of Electrical and Electronics Engineers Inc., pp. 257-260.

Kho, N. D., 2019. *Content and the Connected Home.* [Online]
Available at: http://www.econtentmag.com/Articles/Editorial/Feature/Content-and-the-Connected-Home-102671.htm

Kirtley, J. & Memmel, S., 2018. *Rewriting the "Book of the Machine": Regulatory and Liability Issues for the Internet of Things,* Minnesota: Minnesota Journal of Law, Science & Technology.

Kirtley, J. & Memmel, S., 2018. Too Smart for Its Own Good: Addressing the Privacy and Security Challenges of the Internet of Things. *INTERNET LAW,* october, 22(4), pp. 1-33.

Koo, J. H., December 27, 2017. *Dumb Devices Smarten Up, Widening Data Security Enforcement Net.* [Online]
Available at: https://www.bna.com/dumb-devices-smarten-n73014473644/
[Accessed 26 november 2018].

kozlov, d., veijalainen, j. & ali, y., 2012. *Security and privacy threats in IoT architectures.* Belgium, s.n., pp. 256-262.

Miller, B. & Rowe, D., 2012. *A survey SCADA of and critical infrastructure incidents.* s.l., ACM New York, NY, USA ©2012, pp. 51-56 .

Mishra, A., Nadkarni, K. & Patcha, A., 2004. Intrusion detection in wireless ad hoc networks. *IEEE Wireless Communications,* 11(1), pp. 48 - 60.

Mitchell, R. & Chen, I.-R., 2014. A survey of intrusion detection techniques for cyber-physical systems. *ACM Computing Surveys (CSUR),* 46(4).

noura, h., 2016. *Adaptation of Cryptographic Algorithms According to the Applications Requirements and Limitations : Design, Analyze and Lessons Learned,* paris: University of Pierre & Marie Curie.

Office, Q. G. C. I., 2018. *ICT risk matrix.* [Online]
Available at: https://www.qgcio.qld.gov.au/information-on/ict-risk-management/ict-risk-matrix
[Accessed 25 NOVEMBER 2018].

Park, S.-Y.et al., 2017. *PUFSec: Device fingerprint-based security architecture for Internet of Things.* Atlanta, GA, USA, IEEE.

Patel, A., Qassim, Q. & Wills, C., 2010. A survey of intrusion detection and prevention systems. *Information Management & Computer Security,* 18(4), pp. 277-290.

Perrig, A., Stankovic, J. & Wagner, D., 2004. Security in wireless sensor networks. *Communications of the ACM - Wireless sensor networks,* 47(6), pp. 53-57 .

Qiu, S., Xu, G., Ahmad, H. & Wang, L., 2017. A Robust Mutual Authentication Scheme Based on Elliptic Curve Cryptography for Telecare Medical Information Systems. *IEEE Access ,* Volume 6, pp. 7452 - 7463.

Ray, S., Bhunia, S., Jin, Y. & Tehranipoor, M., 2016. *Security validation in IoT space.* Las Vegas, IEEE.

Sardana, A. & Horrow, . S., 2012. *SecurIT '12 Proceedings of the First International Conference on Security of Internet of Things.* Kollam, India, ACM New York, NY, USA ©2012, pp. 200-203.

Sedjelmaci, H., Senouci, S. M. & Taleb, T., 2017. An Accurate Security Game for Low-Resource IoT Devices. *IEEE Transactions on Vehicular Technology,* 66(10), pp. 9381 - 9393.

Shinde, P. S. & Ardhapurkar, S. B., 2016. *Cyber security analysis using vulnerability assessment and penetration testing.* Coimbatore, India, IEEE.

S. L., W. H. & L. D. X., 2014. Internet of Things in Industries: A Survey. *IEEE Transactions on Industrial Informatics,* november, 10(4), pp. 2233 - 2243.

snort, 2019. *What is Snort?.* [Online]
Available at: https://www.snort.org/faq/what-is-snort

Song, Y., 2013. *Security in Internet of Things,* s.l.: School of Information and Communication Technology (ICT).

SpiceWorks, 2016. *2016 IoT Trends: The Devices have LandedHow IT and IoT are learning to peacefully coexist.* [Online]
Available at: https://www.spiceworks.com/marketing/reports/iot-trends/
[Accessed 28 november 2018].

Symantec, 2016. *An Internet of Things Reference Architecture,* s.l.: Symantec.

Tang, A., 2014. Feature: A guide to penetration testing. *Network Security*, 2014(8), pp. 8-11.

Thapliyal, H., Varun, T. & Kumar, S. D., 2017. *Adiabatic Computing Based Low-Power and DPA-Resistant Lightweight Cryptography for IoT Devices.* Bochum, Germany , IEEE.

The Third Industrial Revolution: A Radical New Sharing Economy. 2018. [Film] s.l.: VICE.

Vacca, J. R., 2013. *Computer and Information Security Handbook.* 2nd ed. s.l.:s.n.

VMware, 2019. *vSphere Distributed Switch.* [Online]
Available at: https://www.vmware.com/latam/products/vsphere/distributed-switch.html
[Accessed 4 april 2019].

Wolff, E. et al., 2017. *Regulatory Rules Of The Road For IoT Manufacturers.* [Online]
Available at: https://www.law360.com/articles/946940/regulatory-rules-of-the-road-for-iot-manufacturers
[Accessed 27 november 2018].

Xu, Q., Ren, P., Song, H. & Du, Q., 2016. Security Enhancement for IoT Communications Exposed to Eavesdroppers With Uncertain Locations. *IEEE Access (Volume: 4),* Volume 4, pp. 2840 - 2853.

Yang, X., Li, Z., Geng, Z. & Zhang, H., 2012. *A Multi-layer Security Model for Internet of Things.* Changsha, China, Springer, Berlin, Heidelberg.

Yu, W. & Köse, S., 2017. A Lightweight Masked AES Implementation for Securing IoT Against CPA Attacks. *2934 - 2944,* 64(11), pp. 2934 - 2944.

Zhang, S. et al., 2017. *Physical layer security in IoT: A spatial-temporal perspective.* Nanjing, China , IEEE.

Zhou, B., Egele, M. & Joshi, A., 2017. *High-performance low-energy implementation of cryptographic algorithms on a programmable SoC for IoT devices.* Waltham, MA, USA, IEEE.

Appendix chapter

83 Windows Server 2016 Benchmark

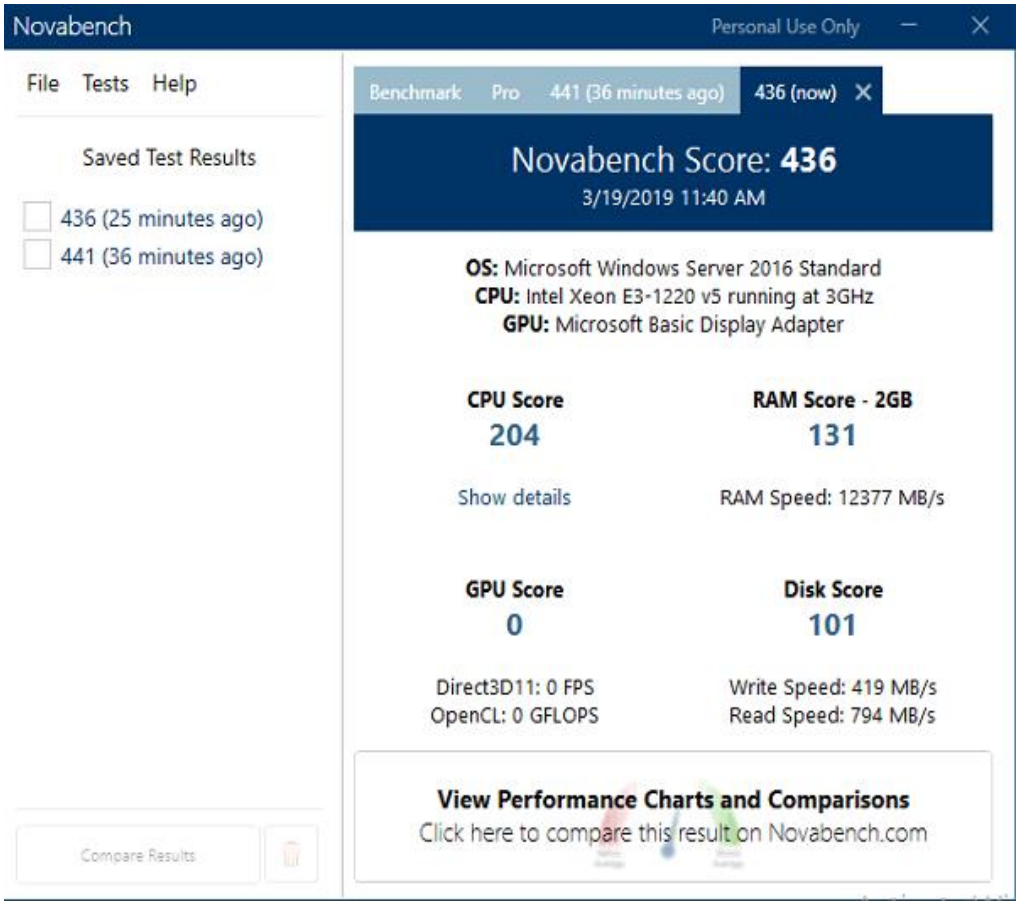


Figure 25 Windows Server 2016 Benchmark

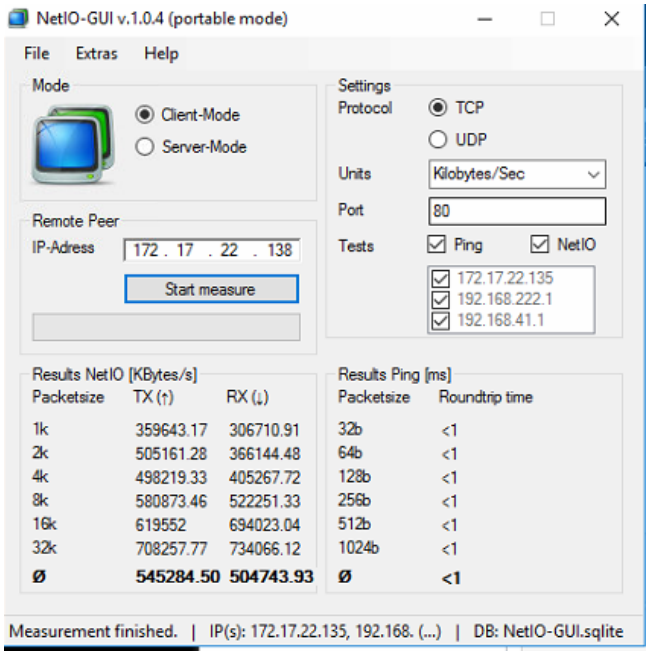


Figure 26 Windows Server 2016 Benchmark

Test Names	Your Computer	Intel(R) Xeon(R) CPU E3-1220 v5 @ 3.00GHz, 3000MHz
Numeric Sort	15.925	14.739
String Sort	68.965	65.882
Bitfield	23.002	22.317
FP Emulation	61.978	61.120
Fourier	24.802	24.585
Assignment	37.610	37.136
IDEA Encryption	34.648	34.276
Huffman	10.585	10.480
Neural NET	30.045	29.758
LU Decomposition	26.976	26.608

INT Overall Score	29.946	29.099
FPU Overall Score	27.190	26.901

Figure 27 Windows Server 2016 Benchmark

84 Kali Linux VM benchmark (Raspberry Pi VM)

CPU

CPU Manufacturer:	GenuineIntel
CPU Type:	Intel(R) Xeon(R) CPU E3-1220 v5 @ 3.00GHz
Speed:	3000.0 MHz
CPUID:	Family 6, Model 94, Stepping 3
Physical CPU's:	1
Cores per CPU:	1
Hyperthreading:	Not capable
CPU Features:	MMX SSE SSE2 SSE3 SSE4.1 SSE4.2 PAE AES AVX AVX2
Cache per CPU package:	
L1 Cache:	16 kB
L2 Cache:	0 kB

Figure 28 Kali Linux VM CPU benchmark (Raspberry Pi VM)

Graphics

VMware SVGA II Adapter (prog-if 00 [VGA controller]	
Memory:	128 MB
Monitor 0:	800x600x24

Figure 29 Kali Linux VM GPU benchmark (Raspberry Pi VM)

Result summary

Test Start time	Thu Apr 25 14:04:03 2019
Test Stop time	Thu Apr 25 14:05:32 2019
Test Duration	000h 01m 29s

Test	Cycles	Operations	Result	Errors	Last Error
CPU - Maths	29	16.786 Billion	PASS	0	No errors
Memory (RAM)	15	1.186 Billion	PASS	0	No errors
2D Graphics	28	2.825 Million	PASS	0	No errors
Disk: Startup Disk [/dev/sda1]	243	73.641 Billion	PASS	0	No errors
Network: eth0 (127.0.0.1)	2045	17.996 Million	PASS	0	No errors
CD/DVD/BD: CD/DVD/BD (/media/cdrom0) [/dev/sr0]	0	0	FAIL	29	Could not determine type of Passmark optical test disk

Figure 30 Kali Linux VM summary benchmark (Raspberry Pi VM)

85 Kali Linux VM benchmark (Banana Pi M64 VM)

CPU

CPU Manufacturer:	GenuineIntel
CPU Type:	Intel(R) Xeon(R) CPU E3-1220 v5 @ 3.00GHz
Speed:	3000.0 MHz
CPUID:	Family 6, Model 94, Stepping 3
Physical CPU's:	1
Cores per CPU:	1
Hyperthreading:	Not capable
CPU Features:	MMX SSE SSE2 SSE3 SSE4.1 SSE4.2 PAE AES AVX AVX2
Cache per CPU package:	
L1 Cache:	16 kB
L2 Cache:	0 kB

Figure 31 Kali Linux VM CPU benchmark (Banana Pi M64 VM)

Graphics

VMware SVGA II Adapter (prog-if 00 [VGA controller]	
Memory:	128 MB
Monitor 0:	800x600x24

Figure 32 Kali Linux VM GPU benchmark (Banana Pi M64 VM)

Result summary

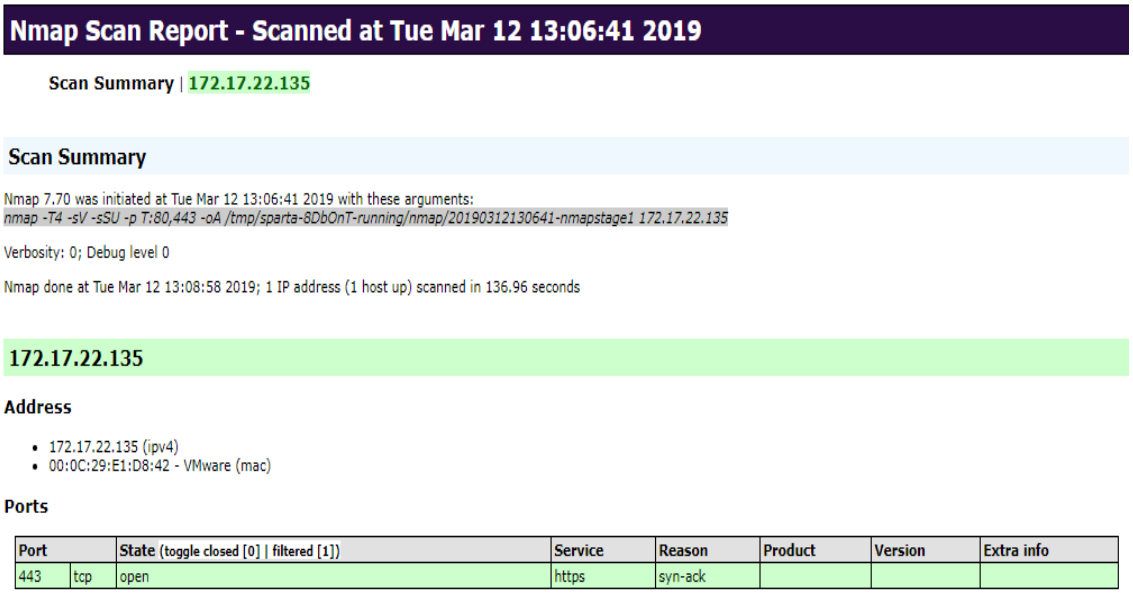
Test Start time	Thu Apr 25 14:04:03 2019
Test Stop time	Thu Apr 25 14:05:32 2019
Test Duration	000h 01m 29s

Test	Cycles	Operations	Result	Errors	Last Error
CPU - Maths	29	16.786 Billion	PASS	0	No errors
Memory (RAM)	15	1.186 Billion	PASS	0	No errors
2D Graphics	28	2.825 Million	PASS	0	No errors
Disk: Startup Disk [/dev/sda1]	243	73.641 Billion	PASS	0	No errors
Network: eth0 (127.0.0.1)	2045	17.996 Million	PASS	0	No errors
CD/DVD/BD: CD/DVD/BD (/media/cdrom0) [/dev/sr0]	0	0	FAIL	29	Could not determine type of Passmark optical test disk

Figure 33 Kali Linux VM summary benchmark (Banana Pi M64 VM)

86 Nmap Benchmark

86.1 Nmap benchmark (Raspberry PI VM)



Nmap Scan Report - Scanned at Tue Mar 12 13:06:41 2019

Scan Summary | 172.17.22.135

Scan Summary

Nmap 7.70 was initiated at Tue Mar 12 13:06:41 2019 with these arguments:
nmap -T4 -sV -sSU -p T:80,443 -oA /tmp/sparta-8DbOnT-running/nmap/20190312130641-nmapstage1 172.17.22.135

Verbosity: 0; Debug level 0

Nmap done at Tue Mar 12 13:08:58 2019; 1 IP address (1 host up) scanned in 136.96 seconds

172.17.22.135

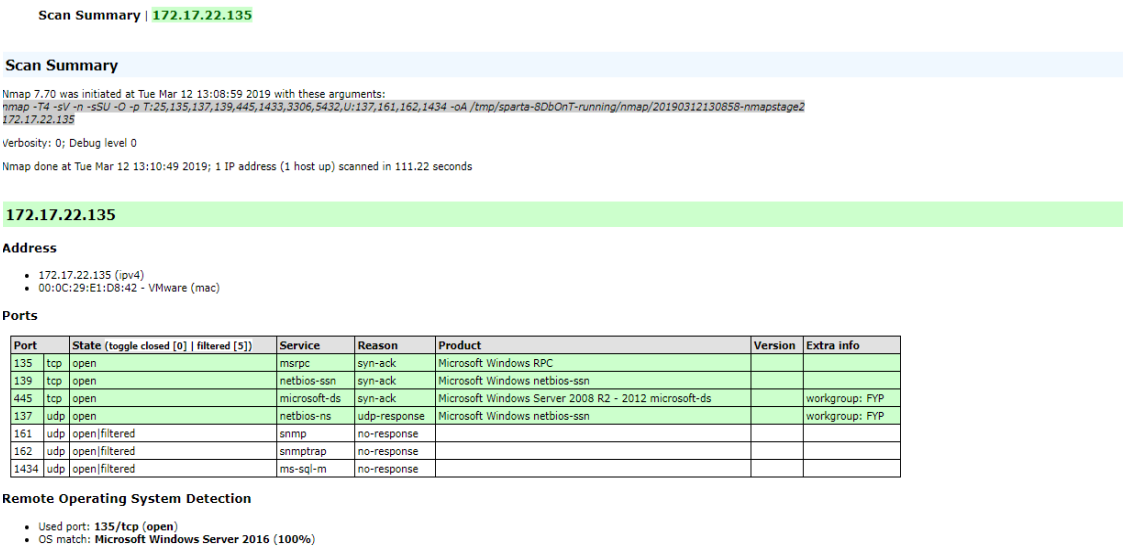
Address

- 172.17.22.135 (ipv4)
- 00:0C:29:E1:D8:42 - VMware (mac)

Ports

Port		State (toggle closed [0] | filtered [1])	Service	Reason	Product	Version	Extra info
443	tcp	open	https	syn-ack			

Figure 34 Nmap benchmark (Raspberry PI VM) stage 1



Scan Summary | 172.17.22.135

Scan Summary

Nmap 7.70 was initiated at Tue Mar 12 13:08:59 2019 with these arguments:
nmap -T4 -sV -n -sSU -O -p T:25,135,137,139,445,1433,3306,5432,U:137,161,162,1434 -oA /tmp/sparta-8DbOnT-running/nmap/20190312130858-nmapstage2 172.17.22.135

Verbosity: 0; Debug level 0

Nmap done at Tue Mar 12 13:10:49 2019; 1 IP address (1 host up) scanned in 111.22 seconds

172.17.22.135

Address

- 172.17.22.135 (ipv4)
- 00:0C:29:E1:D8:42 - VMware (mac)

Ports

Port		State (toggle closed [0] | filtered [5])	Service	Reason	Product	Version	Extra info
135	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
139	tcp	open	netbios-ssn	syn-ack	Microsoft Windows netbios-ssn		
445	tcp	open	microsoft-ds	syn-ack	Microsoft Windows Server 2008 R2 - 2012 microsoft-ds		workgroup: FYP
137	udp	open	netbios-ns	udp-response	Microsoft Windows netbios-ssn		workgroup: FYP
161	udp	open|filtered	snmp	no-response			
162	udp	open|filtered	snmptrap	no-response			
1434	udp	open|filtered	ms-sql-m	no-response			

Remote Operating System Detection

- Used port: **135/tcp** (**open**)
- OS match: **Microsoft Windows Server 2016** (**100%**)

Figure 35 Nmap benchmark (Raspberry PI VM) stage 2

Nmap Scan Report - Scanned at Tue Mar 12 13:10:50 2019

Scan Summary | **172.17.22.135**

Scan Summary

Nmap 7.70 was initiated at Tue Mar 12 13:10:50 2019 with these arguments:
nmap -T4 -sV -n -sSU -p T:23,21,22,110,111,2049,3389,8080,U:500,5060 -oA /tmp/sparta-8DbOnT-running/nmap/20190312131050-nmapstage3 172.17.22.135

Verbosity: 0; Debug level 0

Nmap done at Tue Mar 12 13:12:40 2019; 1 IP address (1 host up) scanned in 110.61 seconds

172.17.22.135

Address

- 172.17.22.135 (ipv4)
- 00:0C:29:E1:D8:42 - VMware (mac)

Ports

Port		State (toggle closed [0] | filtered [7])	Service	Reason	Product	Version	Extra info
3389	tcp	open	ms-wbt-server	syn-ack	Microsoft Terminal Services		
500	udp	open|filtered	isakmp	no-response			
5060	udp	open|filtered	sip	no-response			

Figure 36 Nmap benchmark (Raspberry PI VM) stage 3

Scan Summary

Nmap 7.70 was initiated at Tue Mar 12 13:12:40 2019 with these arguments:
nmap -T4 -sV -n -sSU -p T:0-20,24,26-79,81-109,112-134,136,138,140-442,444,446-1432,1434-2048,2050-3305,3307-3388,3390-5431,5433-8079,8081-29999 -oA /tmp/sparta-8DbOnT-running/nmap/20190312131240-nmapstage4 172.17.22.135

Verbosity: 0; Debug level 0

Nmap done at Tue Mar 12 13:15:49 2019; 1 IP address (1 host up) scanned in 188.46 seconds

172.17.22.135

Address

- 172.17.22.135 (ipv4)
- 00:0C:29:E1:D8:42 - VMware (mac)

Ports

The 29970 ports scanned but not shown below are in state: **filtered**

- 29970 ports replied with: **no-responses**

Port		State (toggle closed [0] | filtered [0])	Service	Reason	Product	Version	Extra info
53	tcp	open	domain	syn-ack			
88	tcp	open	kerberos-sec	syn-ack	Microsoft Windows Kerberos		server time: 2019-03-13 05:13:31Z
389	tcp	open	ldap	syn-ack	Microsoft Windows Active Directory LDAP		Domain: fyp.server.com, Site: Default-First-Site-Name
464	tcp	open	kpasswd5	syn-ack			
593	tcp	open	ncacn_http	syn-ack	Microsoft Windows RPC over HTTP	1.0	
636	tcp	open	tcpwrapped	syn-ack			
902	tcp	open	vmware-auth	syn-ack	VMware Authentication Daemon	1.10	Uses VNC, SOAP
912	tcp	open	vmware-auth	syn-ack	VMware Authentication Daemon	1.0	Uses VNC, SOAP
3268	tcp	open	ldap	syn-ack	Microsoft Windows Active Directory LDAP		Domain: fyp.server.com, Site: Default-First-Site-Name
3269	tcp	open	tcpwrapped	syn-ack			
5985	tcp	open	http	syn-ack	Microsoft HTTPAPI httpd	2.0	SSDP/UPnP
9389	tcp	open	mc-nmf	syn-ack	.NET Message Framing		

Figure 37 Nmap benchmark (Raspberry PI VM) stage 4

86.2 Nmap benchmark (Banana PI VM)

Nmap Scan Report - Scanned at Tue Mar 12 13:06:41 2019

Scan Summary | 172.17.22.135

Scan Summary

Nmap 7.70 was initiated at Tue Mar 12 13:06:41 2019 with these arguments:
nmap -T4 -sV -sSU -p T:80,443 -oA /tmp/sparta-8DbOnT-running/nmap/20190312130641-nmapstage1 172.17.22.135

Verbosity: 0; Debug level 0

Nmap done at Tue Mar 12 13:08:58 2019; 1 IP address (1 host up) scanned in 136.96 seconds

172.17.22.135

Address

- 172.17.22.135 (ipv4)
- 00:0C:29:E1:D8:42 - VMware (mac)

Ports

Port		State (toggle closed [0] | filtered [1])	Service	Reason	Product	Version	Extra info
443	tcp	open	https	syn-ack			

Figure 38 Nmap benchmark (Banana PI VM) stage 1

Scan Summary | 172.17.22.135

Scan Summary

Nmap 7.70 was initiated at Tue Mar 12 13:08:59 2019 with these arguments:
nmap -T4 -sV -n -sSU -O -p T:25,135,137,139,445,1433,3306,5432,U:137,161,162,1434 -oA /tmp/sparta-8DbOnT-running/nmap/20190312130858-nmapstage2 172.17.22.135

Verbosity: 0; Debug level 0

Nmap done at Tue Mar 12 13:10:49 2019; 1 IP address (1 host up) scanned in 111.22 seconds

172.17.22.135

Address

- 172.17.22.135 (ipv4)
- 00:0C:29:E1:D8:42 - VMware (mac)

Ports

Port		State (toggle closed [0] | filtered [5])	Service	Reason	Product	Version	Extra info
135	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
139	tcp	open	netbios-ssn	syn-ack	Microsoft Windows netbios-ssn		
445	tcp	open	microsoft-ds	syn-ack	Microsoft Windows Server 2008 R2 - 2012 microsoft-ds		workgroup: FYP
137	udp	open	netbios-ns	udp-response	Microsoft Windows netbios-ssn		workgroup: FYP
161	udp	open|filtered	snmp	no-response			
162	udp	open|filtered	snmptrap	no-response			
1434	udp	open|filtered	ms-sql-m	no-response			

Remote Operating System Detection

- Used port: **135/tcp** (**open**)
- OS match: **Microsoft Windows Server 2016** (**100%**)

Figure 39 Nmap benchmark (Banana PI VM) stage 2

Nmap Scan Report - Scanned at Tue Mar 12 13:10:50 2019

Scan Summary | **172.17.22.135**

Scan Summary

Nmap 7.70 was initiated at Tue Mar 12 13:10:50 2019 with these arguments:
nmap -T4 -sV -n -sSU -p T:23,21,22,110,111,2049,3389,8080,U:500,5060 -oA /tmp/sparta-8DbOnT-running/nmap/20190312131050-nmapstage3 172.17.22.135

Verbosity: 0; Debug level 0

Nmap done at Tue Mar 12 13:12:40 2019; 1 IP address (1 host up) scanned in 110.61 seconds

172.17.22.135

Address

- 172.17.22.135 (ipv4)
- 00:0C:29:E1:D8:42 - VMware (mac)

Ports

Port		State (toggle closed [0] | filtered [7])	Service	Reason	Product	Version	Extra info
3389	tcp	open	ms-wbt-server	syn-ack	Microsoft Terminal Services		
500	udp	open|filtered	isakmp	no-response			
5060	udp	open|filtered	sip	no-response			

Figure 40 Nmap benchmark (Banana PI VM) stage 3

Scan Summary

Nmap 7.70 was initiated at Tue Mar 12 13:12:40 2019 with these arguments:
nmap -T4 -sV -n -sSU -p T:0-20,24,26-79,81-109,112-134,136,138,140-442,444,446-1432,1434-2048,2050-3305,3307-3388,3390-5431,5433-8079,8081-29999 -oA /tmp/sparta-8DbOnT-running/nmap/20190312131240-nmapstage4 172.17.22.135

Verbosity: 0; Debug level 0

Nmap done at Tue Mar 12 13:15:49 2019; 1 IP address (1 host up) scanned in 188.46 seconds

172.17.22.135

Address

- 172.17.22.135 (ipv4)
- 00:0C:29:E1:D8:42 - VMware (mac)

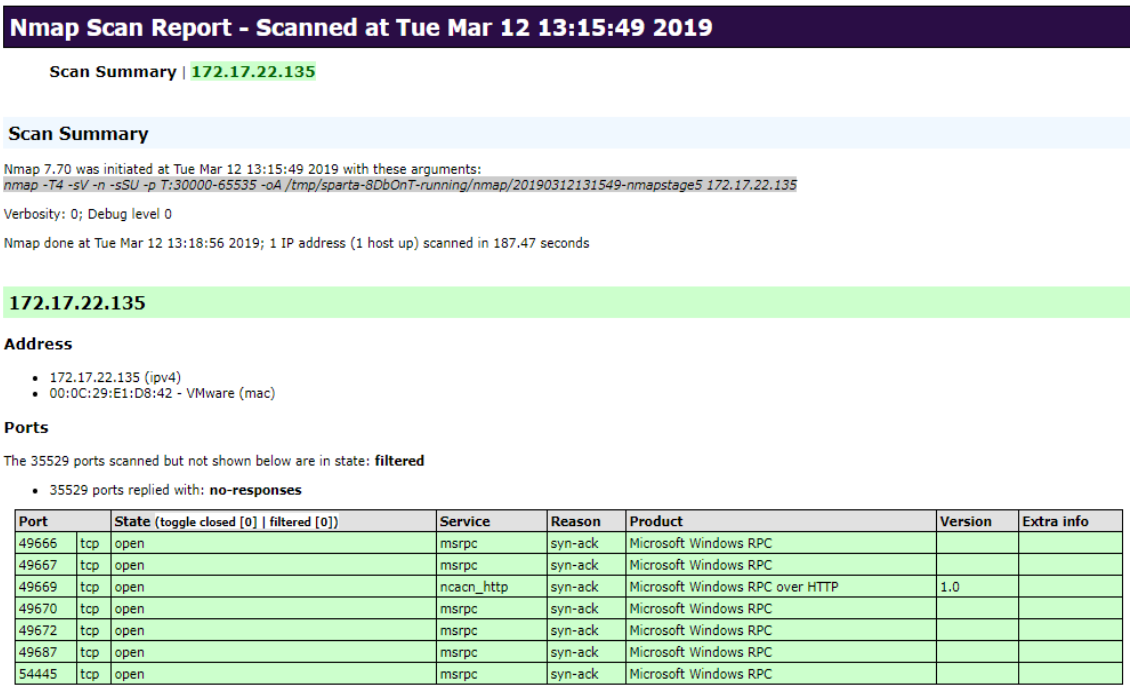
Ports

The 29970 ports scanned but not shown below are in state: **filtered**

- 29970 ports replied with: **no-responses**

Port		State (toggle closed [0] | filtered [0])	Service	Reason	Product	Version	Extra info
53	tcp	open	domain	syn-ack			
88	tcp	open	kerberos-sec	syn-ack	Microsoft Windows Kerberos		server time: 2019-03-13 05:13:31Z
389	tcp	open	ldap	syn-ack	Microsoft Windows Active Directory LDAP		Domain: fyp.server.com, Site: Default-First-Site-Name
464	tcp	open	kpasswd5	syn-ack			
593	tcp	open	ncacn_http	syn-ack	Microsoft Windows RPC over HTTP	1.0	
636	tcp	open	tcpwrapped	syn-ack			
902	tcp	open	vmware-auth	syn-ack	VMware Authentication Daemon	1.10	Uses VNC, SOAP
912	tcp	open	vmware-auth	syn-ack	VMware Authentication Daemon	1.0	Uses VNC, SOAP
3268	tcp	open	ldap	syn-ack	Microsoft Windows Active Directory LDAP		Domain: fyp.server.com, Site: Default-First-Site-Name
3269	tcp	open	tcpwrapped	syn-ack			
5985	tcp	open	http	syn-ack	Microsoft HTTPAPI httpd	2.0	SSDP/UPnP
9389	tcp	open	mc-nmf	syn-ack	.NET Message Framing		

Figure 41Nmap benchmark (Banana PI VM) stage 4



Nmap Scan Report - Scanned at Tue Mar 12 13:15:49 2019

Scan Summary | **172.17.22.135**

Scan Summary

Nmap 7.70 was initiated at Tue Mar 12 13:15:49 2019 with these arguments:
nmap -T4 -sV -n -sSU -p T:30000-65535 -oA /tmp/sparta-8DbOnT-running/nmap/20190312131549-nmapstage5 172.17.22.135

Verbosity: 0; Debug level 0

Nmap done at Tue Mar 12 13:18:56 2019; 1 IP address (1 host up) scanned in 187.47 seconds

172.17.22.135

Address

- 172.17.22.135 (ipv4)
- 00:0C:29:E1:D8:42 - VMware (mac)

Ports

The 35529 ports scanned but not shown below are in state: **filtered**

- 35529 ports replied with: **no-responses**

Port		State (toggle closed [0] | filtered [0])	Service	Reason	Product	Version	Extra info
49666	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
49667	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
49669	tcp	open	ncacn_http	syn-ack	Microsoft Windows RPC over HTTP	1.0	
49670	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
49672	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
49687	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
54445	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		

Figure 42 Nmap benchmark (Banana PI VM) stage 5

87 SPARTA Benchmark

88 SPARTA benchmark (Raspberry PI VM)

Host is up (0.00014s latency).

PORT STATE SERVICE VERSION

21/tcp filtered ftp

22/tcp filtered ssh

23/tcp filtered telnet

110/tcp filtered pop3

111/tcp filtered rpcbind

2049/tcp filtered nfs

3389/tcp open ms-wbt-server Microsoft Terminal Services

8080/tcp filtered http-proxy

500/udp open|filtered isakmp

5060/udp open|filtered sip

MAC Address: 00:0C:29:E1:D8:42 (VMware)

Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .

Nmap done: 1 IP address (1 host up) scanned in 110.61 seconds

---

———————————’- Nikto v2.1.6

+ Target IP: 172.17.22.135

+ Target Hostname: 172.17.22.135

+ Target Port: 443

+ SSL Info: Subject: /C=US/L=Palo Alto/OU=VMware/CN=VMware/emailAddress=none@vmware.com

Ciphers: ECDHE-RSA-AES256-GCM-SHA384

Issuer: /C=US/L=Palo Alto/OU=VMware/CN=VMware/emailAddress=none@vmware.com

+ Start Time: 2019-03-12 13:08:59 (GMT-4)

+ Server: No banner retrieved

+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS

+ No CGI Directories found (use '-C all' to force check all possible dirs)

+ Hostname '172.17.22.135' does not match certificate's names: VMware

+ 7535 requests: 0 error(s) and 2 item(s) reported on remote host

+ End Time: 2019-03-12 13:10:27 (GMT-4) (88 seconds)

+ 1 host(s) tested————————————Š”+Starting Nmap 7.70 (https://nmap.org) at 2019-03-12 13:08 EDT

Nmap scan report for 172.17.22.135

Host is up (0.000093s latency).

PORT STATE SERVICE VERSION

25/tcp filtered smtp

135/tcp open msrpc Microsoft Windows RPC

137/tcp filtered netbios-ns

139/tcp open netbios-ssn Microsoft Windows netbios-ssn

445/tcp open microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds (workgroup: FYP)

1433/tcp filtered ms-sql-s

3306/tcp filtered mysql

5432/tcp filtered postgresql

137/udp open netbios-ns Microsoft Windows netbios-ssn (workgroup: FYP)

161/udp open|filtered snmp

162/udp open|filtered snmptrap

1434/udp open|filtered ms-sql-m

MAC Address: 00:0C:29:E1:D8:42 (VMware)

Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port

Device type: general purpose

Running: Microsoft Windows 2016

OS CPE: cpe:/o:microsoft:windows_server_2016

OS details: Microsoft Windows Server 2016

Network Distance: 1 hop

Service Info: Host: MAINSERVER2016; OS: Windows; CPE: cpe:/o:microsoft:windows

Doing NBT name scan for addresses from 172.17.22.135

NetBIOS Name Table for Host 172.17.22.135:

Incomplete packet, 191 bytes long.

Name Service Type

MAINSERVER2016 Workstation Service

FYP Domain Name

FYP Domain Controllers

MAINSERVER2016 File Server Service

FYP Domain Master Browser

Adapter address: 00:0c:29:e1:d8:42

ORT STATE SERVICE

139/tcp open netbios-ssn

445/tcp open microsoft-ds

MAC Address: 00:0C:29:E1:D8:42 (VMware)

Host script results:

| smb-enum-shares:

| note: ERROR: Enumerating shares failed, guessing at common ones (NT_STATUS_ACCESS_DENIED)

| account_used: <blank>

| \\172.17.22.135\ADMIN$:

| warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED

| Anonymous access: <none>

| \\172.17.22.135\C$:

| warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED

| Anonymous access: <none>

| \\172.17.22.135\IPC$:

| warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED

| Anonymous access: READ

| \\172.17.22.135\NETLOGON:

| warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED

|_ Anonymous access: <none>

Not shown: 35529 filtered ports

PORT STATE SERVICE VERSION

49666/tcp open msrpc Microsoft Windows RPC

49667/tcp open msrpc Microsoft Windows RPC

49669/tcp open ncacn_http Microsoft Windows RPC over HTTP 1.0

49670/tcp open msrpc Microsoft Windows RPC

49672/tcp open msrpc Microsoft Windows RPC

49687/tcp open msrpc Microsoft Windows RPC

54445/tcp open msrpc Microsoft Windows RPC

MAC Address: 00:0C:29:E1:D8:42 (VMware)

Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Target IP: 172.17.22.135

+ Target Hostname: 172.17.22.135

+ Target Port: 5985

+ Start Time: 2019-03-12 13:15:50 (GMT-4)

+ Server: Microsoft-HTTPAPI/2.0

+ The anti-clickjacking X-Frame-Options header is not present.

+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS

+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type

+ No CGI Directories found (use '-C all' to force check all possible dirs)

+ 7536 requests: 0 error(s) and 3 item(s) reported on remote host

+ End Time: 2019-03-12 13:16:01 (GMT-4) (11 seconds)

+ 1 host(s) tested

Not shown: 29970 filtered ports

PORT STATE SERVICE VERSION

53/tcp open domain?

88/tcp open kerberos-sec Microsoft Windows Kerberos (server time: 2019-03-13 05:13:31Z)

389/tcp open ldap Microsoft Windows Active Directory LDAP (Domain: fyp.server.com, Site: Default-First-Site-Name)

464/tcp open kpasswd5?

593/tcp open ncacn_http Microsoft Windows RPC over HTTP 1.0

636/tcp open tcpwrapped

902/tcp open ssl/vmware-auth VMware Authentication Daemon 1.10 (Uses VNC, SOAP)

912/tcp open vmware-auth VMware Authentication Daemon 1.0 (Uses VNC, SOAP)

3268/tcp open ldap Microsoft Windows Active Directory LDAP (Domain: fyp.server.com, Site: Default-First-Site-Name)

3269/tcp open tcpwrapped

5985/tcp open http Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)

9389/tcp open mc-nmf .NET Message Framing

1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :

SF-Port53-TCP:V=7.70%I=7%D=3/12%Time=5C87E8C7%P=x86_64-pc-linux-gnu%r(DNSV

SF:ersionBindReqTCP,20,"\0\x1e\0\x06\x81\x04\0\x01\0\0\0\0\0\0\x07version\

SF:x04bind\0\0\x10\0\x03");

MAC Address: 00:0C:29:E1:D8:42 (VMware)

Service Info: Host: MAINSERVER2016; OS: Windows; CPE: cpe:/o:microsoft:windows

88.1 SPARTA benchmark (Banana PI VM)

mc-nmf.NET Message Framing.! ;
httpMicrosoft HTTPAPI httpd2.0SSDP/UPnP? #E)
vmware-authVMware Authentication Daemon1.0Uses VNC, SOAP@#E)
vmware-authVMware Authentication Daemon1.10Uses VNC, SOAP-!
tcpwrapped3!K
ncacn_httpMicrosoft Windows RPC over HTTP1.0
kpasswd5g [w
ldapMicrosoft Windows Active Directory LDAPDomain: fyp.server.com, Site: Default-First-Site-NameN%A
O
kerberos-secMicrosoft Windows Kerberosserver time: 2019-03-13 05:13:31Z• 7

c55• GTrue13094unicornscan-full-udpunicornscan-full-udp172.17.22.135unicornscan -mU -Ir 1000 172.17.22.135:a -v14 Mar 2019 09:13:0414 Mar 2019 09:14:44/tmp/sparta-

8DbOnT-running/unicornscanfulludp/20190314091304-unicornscanfulludp-172.17.22.135FinishedFalse,0 -'• m55• True18564niktonikto (5985/tcp)172.17.22.1355985tcpnikto -o "/tmp/sparta-8DbOnT-running/nikto/20190312131549-nikto-172.17.22.135-5985.txt" -p 5985 -h 172.17.22.13512 Mar 2019 13:15:4912 Mar 2019 13:16:01/tmp/sparta-8DbOnT-running/nikto/20190312131549-nikto-172.17.22.135-5985FinishedFalse• 0 '7'

PORT STATE SERVICE VERSION

21/tcp filtered ftp

22/tcp filtered ssh

23/tcp filtered telnet

110/tcp filtered pop3

111/tcp filtered rpcbind

2049/tcp filtered nfs

3389/tcp open ms-wbt-server Microsoft Terminal Services

8080/tcp filtered http-proxy

500/udp open|filtered isakmp

5060/udp open|filtered sip

MAC Address: 00:0C:29:E1:D8:42 (VMware)

Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .

Nmap done: 1 IP address (1 host up) scanned in 110.61 seconds

---

‰———————————————’- Nikto v2.1.6

+ Target IP: 172.17.22.135

+ Target Hostname: 172.17.22.135

+ Target Port: 443

+ SSL Info: Subject: /C=US/L=Palo Alto/OU=VMware/CN=VMware/emailAddress=none@vmware.com

Ciphers: ECDHE-RSA-AES256-GCM-SHA384

Issuer: /C=US/L=Palo Alto/OU=VMware/CN=VMware/emailAddress=none@vmware.com

+ Start Time: 2019-03-12 13:08:59 (GMT-4)

+ Server: No banner retrieved

+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS

+ No CGI Directories found (use '-C all' to force check all possible dirs)

+ Hostname '172.17.22.135' does not match certificate's names: VMware

+ 7535 requests: 0 error(s) and 2 item(s) reported on remote host

+ End Time: 2019-03-12 13:10:27 (GMT-4) (88 seconds)

+ 1 host(s) tested———————————————Š”+Starting Nmap 7.70 (https://nmap.org) at 2019-03-12 13:08 EDT

Nmap scan report for 172.17.22.135

Host is up (0.000093s latency).

PORT STATE SERVICE VERSION

25/tcp filtered smtp

135/tcp open msrpc Microsoft Windows RPC

137/tcp filtered netbios-ns

139/tcp open netbios-ssn Microsoft Windows netbios-ssn

445/tcp open microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds (workgroup: FYP)

1433/tcp filtered ms-sql-s

3306/tcp filtered mysql

5432/tcp filtered postgresql

137/udp open netbios-ns Microsoft Windows netbios-ssn (workgroup: FYP)

161/udp open|filtered snmp

162/udp open|filtered snmptrap

1434/udp open|filtered ms-sql-m

MAC Address: 00:0C:29:E1:D8:42 (VMware)

Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port

Device type: general purpose

Running: Microsoft Windows 2016

OS CPE: cpe:/o:microsoft:windows_server_2016

OS details: Microsoft Windows Server 2016

Network Distance: 1 hop

Service Info: Host: MAINSERVER2016; OS: Windows; CPE: cpe:/o:microsoft:windows

88.2 SPARTA benchmark (Raspberry PI VM)

89 Armitage test

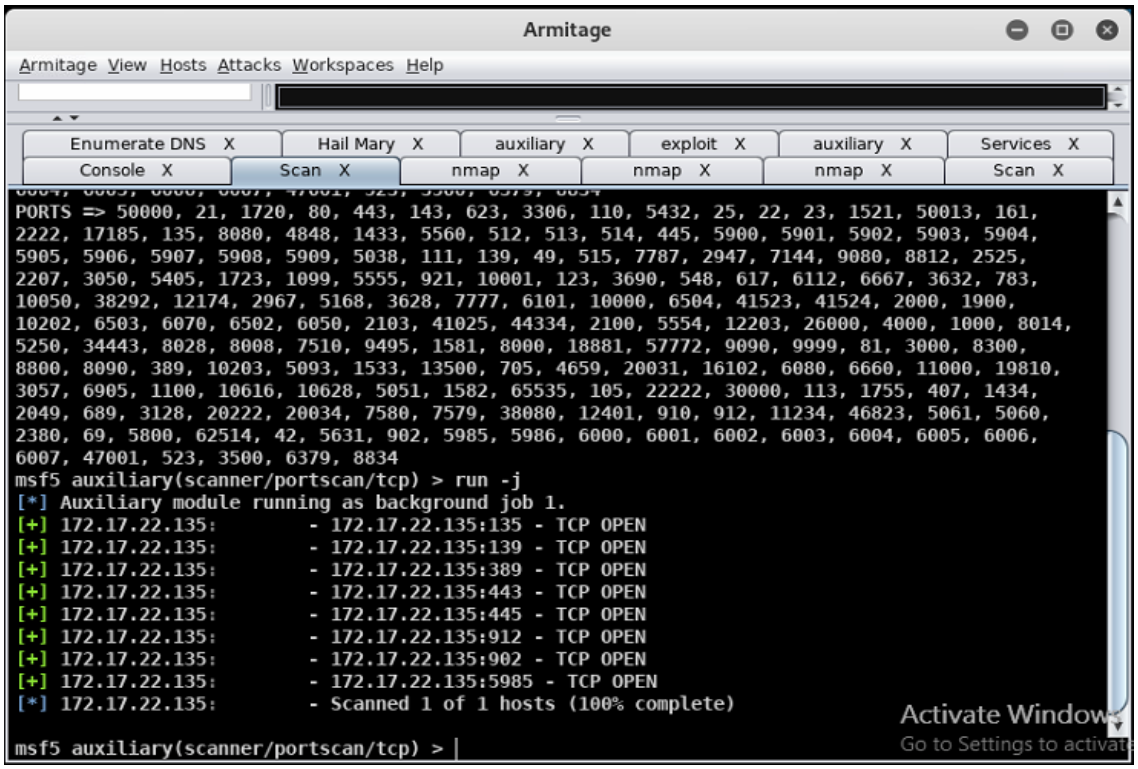


Figure 43 Armitage test scan 1

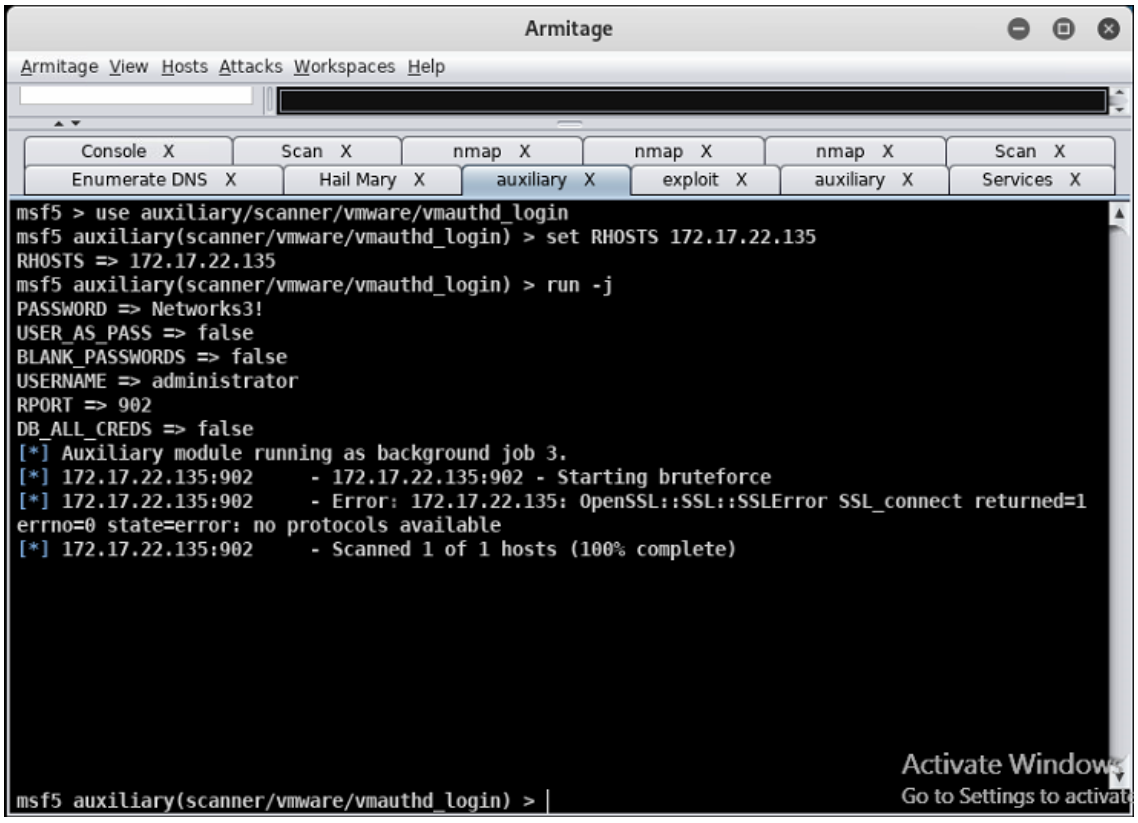


Figure 44 Armitage test scan 2

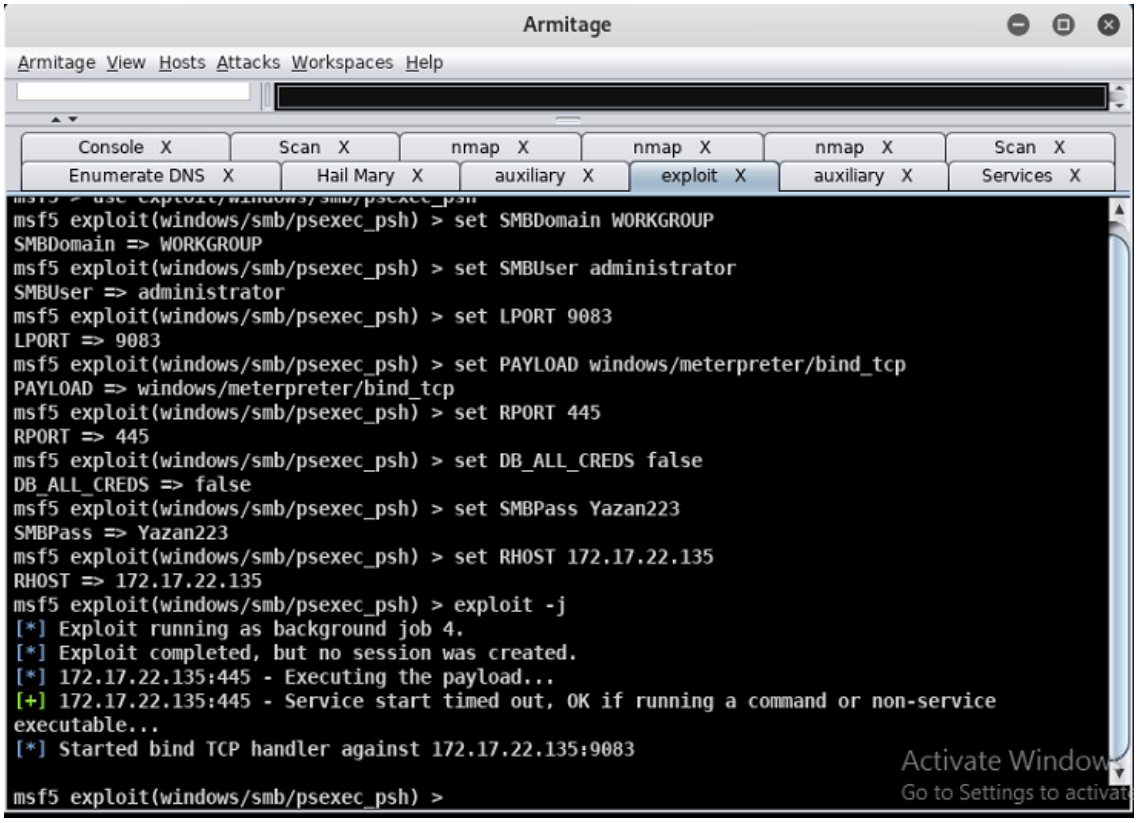


Figure 45 Armitage test scan 3

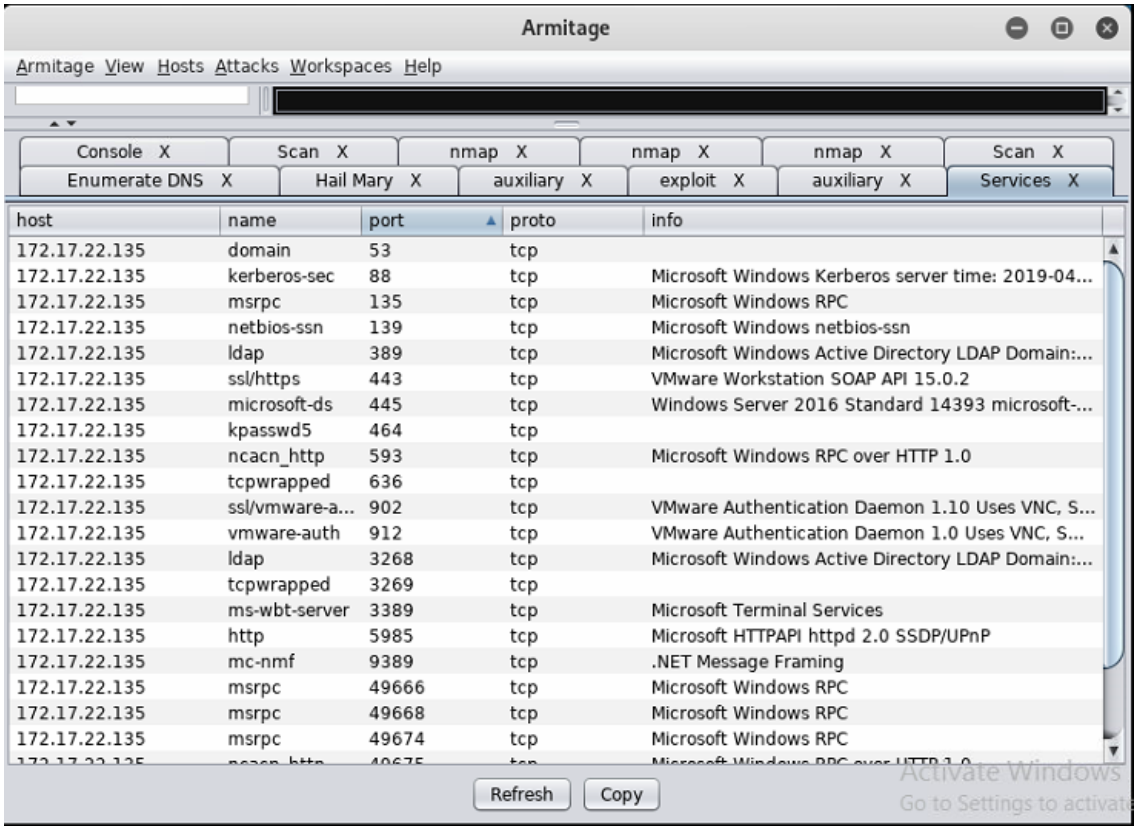


Figure 46 Armitage test scan 4

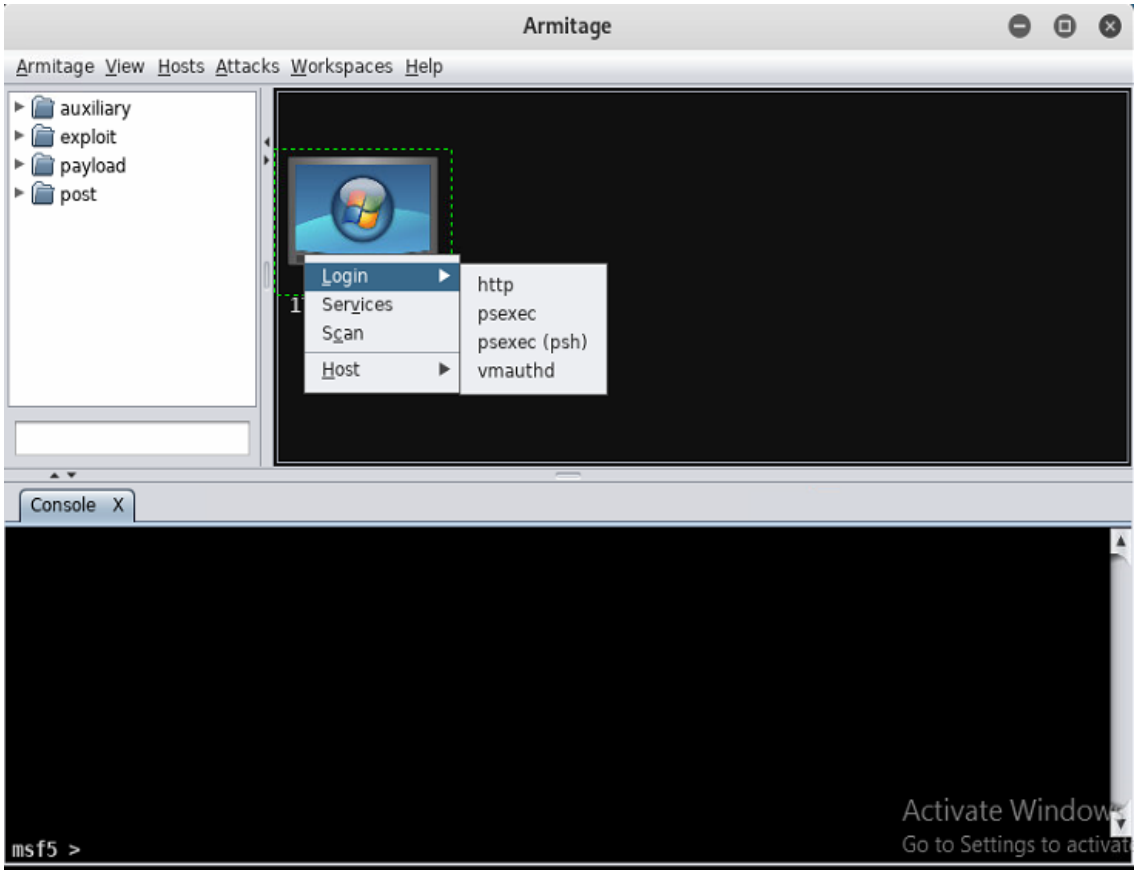


Figure 47 Armitage test scan 5

90 Snort Benchmark (Raspberry PI VM)

Running in IDS mode

--== Initializing Snort ==--

Initializing Output Plugins!

Initializing Preprocessors!

Initializing Plug-ins!

Parsing Rules file "/etc/snort/snort.conf"

PortVar 'HTTP_PORTS' defined : [80:81 311 383 591 593 901 1220 1414 1741 1830 2301 2381 2809 3037 3128 3702 4343 4848 5250 6988 7000:7001 7144:7145 7510 7777 7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180:8181 8243 8280 8300 8800 8888 8899 9000 9060 9080 9090:9091 9443 9999 11371 34443:34444 41080 50002 55555]

PortVar 'SHELLCODE_PORTS' defined : [0:79 81:65535]

PortVar 'ORACLE_PORTS' defined : [1024:65535]

PortVar 'SSH_PORTS' defined : [22]

PortVar 'FTP_PORTS' defined : [21 2100 3535]

PortVar 'SIP_PORTS' defined : [5060:5061 5600]

PortVar 'FILE_DATA_PORTS' defined : [80:81 110 143 311 383 591 593 901 1220 1414 1741 1830 2301 2381 2809 3037 3128 3702 4343 4848 5250 6988 7000:7001 7144:7145 7510 7777 7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180:8181 8243 8280 8300 8800 8888 8899 9000 9060 9080 9090:9091 9443 9999 11371 34443:34444 41080 50002 55555]

PortVar 'GTP_PORTS' defined : [2123 2152 3386]

Detection:

Search-Method = AC-Full-Q

Split Any/Any group = enabled

Search-Method-Optimizations = enabled

Maximum pattern length = 20

Tagged Packet Limit: 256

Loading dynamic engine /usr/lib/snort_dynamicengine/libsf_engine.so... done

Loading all dynamic detection libs from /usr/lib/snort_dynamicrules...

WARNING: No dynamic libraries found in directory /usr/lib/snort_dynamicrules.

Finished Loading all dynamic detection libs from /usr/lib/snort_dynamicrules

Loading all dynamic preprocessor libs from /usr/lib/snort_dynamicpreprocessor/...

Run time for packet processing was 2535.183930 seconds

Snort processed 336290 packets.

Snort ran for 0 days 0 hours 45 minutes 15 seconds

Pkts/min: 8006

Pkts/sec: 132

==========

Memory usage summary:

Total non-mmapped bytes (arena): 50959984

Bytes in mapped regions (hblkhd): 14874144

Total allocated space (uordblks): 40666912

Total free space (fordblks): 9419072

Topmost releasable block (keepcost): 3377336

==========

Packet I/O Totals:

Received: 817037

Analyzed: 336290 (41.160%)
Dropped: 480729 (37.043%)
Filtered: 0 (0.000%)
Outstanding: 480747 (58.840%)
Injected: 0
==========
Breakdown by protocol (includes rebuilt packets):
Eth: 336302 (100.000%)
VLAN: 0 (0.000%)
IP4: 323125 (96.082%)
Frag: 48 (0.014%)
ICMP: 4193 (1.247%)
UDP: 122652 (36.471%)
TCP: 195243 (58.056%)
IP6: 1736 (0.516%)
IP6 Ext: 2260 (0.672%)
IP6 Opts: 524 (0.156%)
Frag6: 0 (0.000%)
ICMP6: 628 (0.187%)
UDP6: 1108 (0.329%)
TCP6: 0 (0.000%)
Teredo: 0 (0.000%)
ICMP-IP: 0 (0.000%)
IP4/IP4: 0 (0.000%)
IP4/IP6: 0 (0.000%)
IP6/IP4: 0 (0.000%)
IP6/IP6: 0 (0.000%)
GRE: 0 (0.000%)
GRE Eth: 0 (0.000%)
GRE VLAN: 0 (0.000%)
GRE IP4: 0 (0.000%)
GRE IP6: 0 (0.000%)
GRE IP6 Ext: 0 (0.000%)

GRE PPTP: 0 (0.000%)
GRE ARP: 0 (0.000%)
GRE IPX: 0 (0.000%)
GRE Loop: 0 (0.000%)
MPLS: 0 (0.000%)
ARP: 9922 (2.950%)
IPX: 0 (0.000%)
Eth Loop: 0 (0.000%)
Eth Disc: 0 (0.000%)
IP4 Disc: 66 (0.020%)
IP6 Disc: 0 (0.000%)
TCP Disc: 0 (0.000%)
UDP Disc: 0 (0.000%)
ICMP Disc: 0 (0.000%)
All Discard: 66 (0.020%)
Other: 2454 (0.730%)
Bad Chk Sum: 2437 (0.725%)
Bad TTL: 0 (0.000%)
S5 G 1: 0 (0.000%)
S5 G 2: 0 (0.000%)
Total: 336302
==========
========
Action Stats:
Alerts: 5353 (1.592%)
Logged: 5353 (1.592%)
Passed: 0 (0.000%)
Limits:
Match: 0
Queue: 0
Log: 0
Event: 0
Alert: 0
Verdicts:

```
      Allow:      336282 ( 41.159%)
      Block:           0 (  0.000%)
    Replace:           0 (  0.000%)
  Whitelist:           8 (  0.001%)
  Blacklist:           0 (  0.000%)
     Ignore:           0 (  0.000%)
      Retry:           0 (  0.000%)
===============================================================================
Frag3 statistics:
       Total Fragments: 48
     Frags Reassembled: 12
             Discards: 0
          Memory Faults: 0
             Timeouts: 0
             Overlaps: 0
            Anomalies: 0
               Alerts: 0
               Drops: 0
    FragTrackers Added: 12
   FragTrackers Dumped: 12
FragTrackers Auto Freed: 0
   Frag Nodes Inserted: 48
    Frag Nodes Deleted: 48
===============================================================================
===============================================================================
Stream statistics:
          Total sessions: 182007
            TCP sessions: 95070
            UDP sessions: 86937
           ICMP sessions: 0
             IP sessions: 0
```

TCP Prunes: 0

UDP Prunes: 0

ICMP Prunes: 0

IP Prunes: 0

TCP StreamTrackers Created: 95251

TCP StreamTrackers Deleted: 95251

TCP Timeouts: 0

TCP Overlaps: 0

TCP Segments Queued: 5

TCP Segments Released: 5

TCP Rebuilt Packets: 0

TCP Segments Used: 0

TCP Discards: 0

TCP Gaps: 0

UDP Sessions Created: 86944

UDP Sessions Deleted: 86944

UDP Timeouts: 7

UDP Discards: 0

Events: 2559

Internal Events: 0

TCP Port Filter

Filtered: 0

Inspected: 0

Tracked: 193035

UDP Port Filter

Filtered: 0

Inspected: 0

Tracked: 86937

==========

==========

SMTP Preprocessor Statistics

Total sessions : 0

Max concurrent sessions : 0

==========

dcerpc2 Preprocessor Statistics

Total sessions: 3

Total sessions aborted: 1

Transports

UDP

Total sessions: 3

Packet stats

Packets: 3

DCE/RPC

Connectionless

Packet stats

Packets: 3

Fragments: 0

Max fragment size: 0

Reassembled: 0

==========

SSL Preprocessor:

SSL packets decoded: 1369

Client Hello: 1

Server Hello: 1

Certificate: 23

Server Done: 91

Client Key Exchange: 0

Server Key Exchange: 1

Change Cipher: 113

Finished: 0

Client Application: 788

Server Application: 1

Alert: 46
Unrecognized records: 509
Completed handshakes: 0
Bad handshakes: 22
Sessions ignored: 1
Detection disabled: 12
==========
SIP Preprocessor Statistics
Total sessions: 20
Total dialogs: 7
Requests: 7
invite: 0
cancel: 0
ack: 0
bye: 0
register: 0
options: 7
refer: 0
subscribe: 0
update: 0
join: 0
info: 0
message: 0
notify: 0
prack: 0
Responses: 0
1xx: 0
2xx: 0
3xx: 0
4xx: 0
5xx: 0
6xx: 0
7xx: 0

8xx: 0

9xx: 0

Ignore sessions: 0

Ignore channels: 0

==========

91 Snort benchmark (Banana Pi VM)

Running in IDS mode

--== Initializing Snort ==--

Initializing Output Plugins!

Initializing Preprocessors!

Initializing Plug-ins!

Parsing Rules file "/etc/snort/snort.conf"

PortVar 'HTTP_PORTS' defined : [80:81 311 383 591 593 901 1220 1414 1741 1830 2301 2381 2809 3037 3128 3702 4343 4848 5250 6988 7000:7001 7144:7145 7510 7777 7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180:8181 8243 8280 8300 8800 8888 8899 9000 9060 9080 9090:9091 9443 9999 11371 34443:34444 41080 50002 55555]

PortVar 'SHELLCODE_PORTS' defined : [0:79 81:65535]

PortVar 'ORACLE_PORTS' defined : [1024:65535]

PortVar 'SSH_PORTS' defined : [22]

PortVar 'FTP_PORTS' defined : [21 2100 3535]

PortVar 'SIP_PORTS' defined : [5060:5061 5600]

PortVar 'FILE_DATA_PORTS' defined : [80:81 110 143 311 383 591 593 901 1220 1414 1741 1830 2301 2381 2809 3037 3128 3702 4343 4848 5250 6988 7000:7001 7144:7145 7510 7777 7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180:8181 8243 8280 8300 8800 8888 8899 9000 9060 9080 9090:9091 9443 9999 11371 34443:34444 41080 50002 55555]

PortVar 'GTP_PORTS' defined : [2123 2152 3386]

Detection:

Search-Method = AC-Full-Q

Split Any/Any group = enabled

Search-Method-Optimizations = enabled

Maximum pattern length = 20

Tagged Packet Limit: 256

Loading dynamic engine /usr/lib/snort_dynamicengine/libsf_engine.so... done

```
Loading all dynamic detection libs from /usr/lib/snort_dynamicrules...
WARNING: No dynamic libraries found in directory /usr/lib/snort_dynamicrules.
  Finished Loading all dynamic detection libs from /usr/lib/snort_dynamicrules
Loading all dynamic preprocessor libs from /usr/lib/snort_dynamicpreprocessor/...

Run time for packet processing was 2535.183930 seconds
Snort processed 336290 packets.
Snort ran for 0 days 0 hours 42 minutes 15 seconds
   Pkts/min:        8006
   Pkts/sec:        132
===============================================================================
Memory usage summary:
  Total non-mmapped bytes (arena):       50089984
  Bytes in mapped regions (hblkhd):      13574144
  Total allocated space (uordblks):      40672912
  Total free space (fordblks):           9417072
  Topmost releasable block (keepcost):   3342336
===============================================================================
Packet I/O Totals:
   Received:      817037
   Analyzed:      336290 ( 41.160%)
    Dropped:      480729 ( 37.043%)
   Filtered:          0 (  0.000%)
Outstanding:      480747 ( 58.840%)
   Injected:          0
===============================================================================
Breakdown by protocol (includes rebuilt packets):
        Eth:      336302 (100.000%)
       VLAN:           0 (  0.000%)
        IP4:      323125 ( 96.082%)
       Frag:          48 (  0.014%)
```

ICMP: 4193 (1.247%)
UDP: 122652 (36.471%)
TCP: 195243 (58.056%)
IP6: 1736 (0.516%)
IP6 Ext: 2260 (0.672%)
IP6 Opts: 524 (0.156%)
Frag6: 0 (0.000%)
ICMP6: 628 (0.187%)
UDP6: 1108 (0.329%)
TCP6: 0 (0.000%)
Teredo: 0 (0.000%)
ICMP-IP: 0 (0.000%)
IP4/IP4: 0 (0.000%)
IP4/IP6: 0 (0.000%)
IP6/IP4: 0 (0.000%)
IP6/IP6: 0 (0.000%)
GRE: 0 (0.000%)
GRE Eth: 0 (0.000%)
GRE VLAN: 0 (0.000%)
GRE IP4: 0 (0.000%)
GRE IP6: 0 (0.000%)
GRE IP6 Ext: 0 (0.000%)
GRE PPTP: 0 (0.000%)
GRE ARP: 0 (0.000%)
GRE IPX: 0 (0.000%)
GRE Loop: 0 (0.000%)
MPLS: 0 (0.000%)
ARP: 9922 (2.950%)
IPX: 0 (0.000%)
Eth Loop: 0 (0.000%)
Eth Disc: 0 (0.000%)
IP4 Disc: 66 (0.020%)
IP6 Disc: 0 (0.000%)
TCP Disc: 0 (0.000%)

UDP Disc: 0 (0.000%)
ICMP Disc: 0 (0.000%)
All Discard: 66 (0.020%)
Other: 2454 (0.730%)
Bad Chk Sum: 2437 (0.725%)
Bad TTL: 0 (0.000%)
S5 G 1: 0 (0.000%)
S5 G 2: 0 (0.000%)
Total: 336302
==========
Action Stats:
Alerts: 5353 (1.592%)
Logged: 5353 (1.592%)
Passed: 0 (0.000%)
Limits:
Match: 0
Queue: 0
Log: 0
Event: 0
Alert: 0
Verdicts:
Allow: 336282 (41.159%)
Block: 0 (0.000%)
Replace: 0 (0.000%)
Whitelist: 8 (0.001%)
Blacklist: 0 (0.000%)
Ignore: 0 (0.000%)
Retry: 0 (0.000%)
==========
Frag3 statistics:
Total Fragments: 48
Frags Reassembled: 12

Discards: 0
Memory Faults: 0
Timeouts: 0
Overlaps: 0
Anomalies: 0
Alerts: 0
Drops: 0
FragTrackers Added: 12
FragTrackers Dumped: 12
FragTrackers Auto Freed: 0
Frag Nodes Inserted: 48
Frag Nodes Deleted: 48
==========
==========
==========
==========
Stream statistics:
Total sessions: 182007
TCP sessions: 95070
UDP sessions: 86937
ICMP sessions: 0
IP sessions: 0
TCP Prunes: 0
UDP Prunes: 0
ICMP Prunes: 0
IP Prunes: 0
TCP StreamTrackers Created: 95251
TCP StreamTrackers Deleted: 95251
TCP Timeouts: 0
TCP Overlaps: 0
TCP Segments Queued: 5
TCP Segments Released: 5
TCP Rebuilt Packets: 0
TCP Segments Used: 0

TCP Discards: 0

TCP Gaps: 0

UDP Sessions Created: 86944

UDP Sessions Deleted: 86944

UDP Timeouts: 7

UDP Discards: 0

Events: 2559

Internal Events: 0

TCP Port Filter

Filtered: 0

Inspected: 0

Tracked: 193035

UDP Port Filter

Filtered: 0

Inspected: 0

Tracked: 86937

==========

==========

SMTP Preprocessor Statistics

Total sessions : 0

Max concurrent sessions : 0

==========

==========

dcerpc2 Preprocessor Statistics

Total sessions: 3

Total sessions aborted: 1

Transports

UDP

Total sessions: 3

Packet stats

Packets: 3

DCE/RPC

Connectionless

Packet stats

Packets: 3

Fragments: 0

Max fragment size: 0

Reassembled: 0

==========

SSL Preprocessor:

SSL packets decoded: 1369

Client Hello: 1

Server Hello: 1

Certificate: 23

Server Done: 91

Client Key Exchange: 0

Server Key Exchange: 1

Change Cipher: 113

Finished: 0

Client Application: 788

Server Application: 1

Alert: 46

Unrecognized records: 509

Completed handshakes: 0

Bad handshakes: 22

Sessions ignored: 1

Detection disabled: 12

==========

SIP Preprocessor Statistics

Total sessions: 20

Total dialogs: 7

Requests: 7

invite: 0
cancel: 0
ack: 0
bye: 0
register: 0
options: 7
refer: 0
subscribe: 0
update: 0
join: 0
info: 0
message: 0
notify: 0
prack: 0
Responses: 0
1xx: 0
2xx: 0
3xx: 0
4xx: 0
5xx: 0
6xx: 0
7xx: 0
8xx: 0
9xx: 0
Ignore sessions: 0
Ignore channels: 0
==========