ETRO VUB-DEPARTMENT OF ELECTRONICS AND INFORMATICS

VRIJE UNIVERSITEIT BRUSSEL

Secure Hypervisor Virtualization Solution for Medical Devices Applications

Operating Systems and Security
2019/2020

Fauziah Permatasari (0566100)
fauziah.permatasari@vub.be
fauziah.permatasari@uni-weimar.de

Prof. Dr. Ir. Martin Timmerman
Mingyang Zhang

Abstract

The current climate of cybersecurity rapidly increases the risk of technology usage. Billions of people are interacting as well as getting assistance from technology in a day-to-day basis. We use passwords in our smartphones, set up our privacy settings in the social media, use strong password combination for our e-mails. But we often neglect the security of our Internet of Things (IoT) and embedded devices. Perhaps this carefreeness is because getting our smart home assistant hacked will only cause a small hit on our wallet. Additionally, we too, forget that medical applications are also embedded devices, and that they are also vulnerable to malicious cyber-attacks. And unlike the previously mentioned cases, this risk is a matter of life. This subject is further highlighted by the recent EU General Protection Data Regulation (GDPR), as well as the soon-to-be implemented Medical Device Regulation (MDR); proving how critical it is to provide secure software and hardware environment for medical devices applications. One of the well-known good software security practice is implementing Virtualization. Which is unfortunately, is still uncommon to be implemented on medical devices applications. Therefore, this project compares performances of different virtualization solutions for the most used operating system on medical devices: Windows OS, with a realistic test plan and penetration test. These experiments will then provide quantitative measurements that can assist in choosing a suitable virtualization solution for a medical devices application.

Table of Contents

1. Introduction

We would expect that medical devices, especially medical assistive devices, would have one of the most secure operating systems implementation. But often, the truth hurts. And if we do not take immediate action right now, it may literally hurt us.

In this introduction, the author will describe the current emergency in the medical devices application, then the biggest issues often faced in the field, and last but not least is introduction to the solution proposed by the author for medical device firms to use as a secure hypervisor virtualization guideline.

1.1. Current State of Medical Devices Operating Systems

It is very concerning to find out that many medical devices to this date still use operating systems which no longer provide support or patch updates [1] [2] [3] [4] [5].

Many of these devices also did not work on hypervisors, meaning that they solely believe in their users' ability (and moral compass). It is true that no device is secure, however, if there is almost to no effort given to secure devices that are often used to support people's life, then it is a negligence.

1.2. Electromagnetic Interference and Hijacking Issues

There are two major issues that medical device applications face in practice, which are electromagnetic interference (EMI) and recently from cyber attacks. In many cases, companies have to replace or even worse, recall their medical products due to the failure in calibration which causing EMI problems on their devices [6]. These EMIs may cause the medical device to slow down, or even shut down. For the first scenario, it would often cause the patients who are having surgeries or assistive services to get critical complications. However, if the device fully shuts down, it will risk the patients' life [7].

For this problem, in an operating system and software level, what we can do to mitigate it is by implementing hypervisor virtualization [8]. The hypervisor virtual machines as well as the host device shall pass regularly scheduled stress tests that affect its performance and hardware temperature, to show that they can manage to stay up with heavy load of tasks and high machine temperature [9].

Cyber attacks, on the other hand, were not used to be the main priority for medical devices companies. However, in the past years, there are concerning numbers of attacks on medical environments and devices [10]. This can be, in a lower risk spectrum, to have patients' data accessed and publicized involuntarily. And in a higher risk spectrum, is to have risk of life loss from patients who are getting assistance from medical devices [11]. Therefore, nowadays, it is critically important to have a reliable and secure solution towards these medical devices' issues. To avoid this problem, we can do automated vulnerability assessment and penetration testing. This is to make sure that the device is always safe against cyber attacks.

1.3. Proposed Solution Guideline

From the concerns and issues mentioned earlier, the author proposed a guideline to creating a safe, secure, and reliable hypervisor virtualization solution for medical devices applications. This solution includes the monitoring and standardization of medical devices operating system and its virtual machines, as well as vulnerability assessment plan, and an automated penetration testing tool. The solution is catered for specifically medical devices and hospital network environment, however, this solution can also be implemented on other fields' devices. Last but not least, the solution is meant to be fully replicable and available to use without cost, this is to encourage firms and users themselves to be more cyber-aware to protect their patients and of course themselves.

2. Requirement Analysis

There are certain requirement analysis to take in order to comply with the current and upcoming European Council's regulations: General Data Protection (GDPR) and Medical Devices Regulation (MDR) [12]. Additionally, cost and availability of the resources needed for the solution also matter much towards the medical device company.

To achieve this, first, we have to determine what operating systems and security requirements each of these regulations expect. From these considerations, we can decide the suitable standard(s) and variables we shall use as measurements in fulfilling the requirements. Then, secondly, we need to compare available tools and applications that can be used for the complete solution package. And these tools should meet the business and user requirements simultaneously.

2.1. Legal Regulations

Referring to **GDPR's Article 32: Security of processing, Paragraph 1** (European Council, 2016), the operating systems and security's technical responsibility includes:

(a) the pseudonymisation and encryption of personal data;
(b) the ability to ensure the ongoing confidentiality, integrity, availability and resilience of processing systems and services;
(c) the ability to restore the availability and access to personal data in a timely manner in the event of a physical or technical incident;
(d) a process for regularly testing, assessing and evaluating the effectiveness of technical and organisational measures for ensuring the security of the processing.

As for **MDR's Article 10: General obligations of manufacturers, Paragraph 9** (European Council, 2017), the operating systems and security's technical responsibility additionally includes:

(k) processes for reporting of serious incidents and field safety corrective actions in the context of vigilance;
(l) management of corrective and preventive actions and verification of their effectiveness;

Sorting these requirements into a table below, we can find the method on how to accomplish it more clearly.

Requirements	***Solutions***
Article 10 (Paragraph 2) : *Manufacturers shall establish, document, implement and maintain a system for risk management as described in Section 3 of Annex I.*	***1. Hypervisors Standardization and Benchmarking.*** ***2. Vulnerability Assessment & Penetration Testing plan.*** ***3. Recovery Plan.***
Article 10 (Paragraph 9) : *(k) processes for reporting of serious incidents and field safety corrective actions in the context of vigilance;* *(l) management of corrective and preventive actions and verification of their effectiveness;*	
Annex I: *17.4. Manufacturers shall set out minimum requirements concerning hardware, IT networks characteristics and IT security measures, including protection against unauthorised access, necessary to run the software as intended.*	

Table 1. Regulated requirements and proposed solutions.

From these requirements, this project will provide a proposed virtualization and security model for medical devices applications. The project includes reliable hypervirtualization options, management and performance monitoring of the virtual machines, as well as security analysis that conducts vulnerability assessment and penetration testing.

2.2. Business and User Requirements

As mentioned earlier, this solution is catered towards medical devices companies. We know that business and user requirements play a major role on the implementation of such systems. And most of the time, due to inability to adjust to the business goals and users' needs, it may lead to failure of implementation. The next table describes the business requirements for this solution.

Business Requirements
1. Free and open sourced applications. 2. Possible automated and scheduled runs. 3. Easy documentations. 4. Reliable. 5. Resources availability (in case of needing assistance and service).

Table 2. Business requirements for medical device operating system solution.

Meanwhile, for user requirements, we will use two user profiles. These user profiles represent the two level of expertise the medical device users might have: expert (for device IT administrator), and casual (hospital workers that dev) This can be seen on the following tables on the next page.

Expert User Requirements	
1. Comfortable with installing and setting up virtual machines.	Manual installation of virtualizations.
2. Familiar with Windows Operating System set-up and services.	Able to adjust services catalogue.
3. Comfortable with Command Line Interface.	Graphics User Interface is not mandatory.
4. Knowledgable about operating systems architecture.	Know about in-depth analysis of CPU and RAM use.

Table 3. Expert User profile and requirement list.

Casual User Requirements	
1. Uncomfortable with installing and setting up virtual machines	Virtual machines and network should be ready for use.
2. Unfamiliar with automated and scheduled runs or tests.	Tests and monitoring should already be automatically set.
3. Prefer Graphics User Interface and straight-forward output.	Graphics User Interface is mandatory.
4. Basic knowledge about operating systems architecture.	Explicit and straight-forward recovery options.

Table 4. Casual User profile and requirement list.

3. Technical Specifications

In this chapter, there are three technical specifications to be defined. These include; firstly, the actual devices used for the experiments themselves, and next is the medical devices models specifications, then thirdly the hypervisors to be used in each operating system variables, last but not least is the test environment itself. The author referred to real-life technical specifications of well-known medical device products. This was to guarantee that all the technical specifications reasoning for the models were made as close as possible as the real-life implementation of medical devices and healthcare network architecture.

4.1. Device used on the experiments

The device used in this project was a Windows OS PC, and both the native or bare-metal hypervisors as well as the operating system or hosted hypervisors were tested on this device. This is in order to give fair judgement for the benchmarking and performance comparisons between the hypervisors.

	Experiment Device
Model Identifier	Desktop-1LJI3AF
Processor	Intel Core i5
Processor Speed	3.6 GHz
Number of Processors	1
Number of Cores	2
RAM	8 GB
Hyper-threading	Enabled
Operating System	Windows 10 Education
Version	1903
Bit	x64
OS Build	18362.592

Table 5. Experiment device hardware and software specifications.

4.2. Test 'medical devices' specifications

These specifications were replicated as close as possible on to the hypervisors machines used in the experiments. There are two medical device products that were referred as the targeted devices. The first device would be a **Cardio Workstation**, and the second device would be a **Ventricular Assist Device**.

These devices are very commonly used in healthcare, with some of them often make use of virtual machines in their operating system. And unlike embedded medical devices, these devices are solely supported by a full operating system. Therefore, having these two devices as the models are very important to represent medical devices possessing highly critical risks, as any kind of misuse and/or malicious attempts carried out on the devices will result in great damage towards the patients as their users. Whether it is in the form of data leakage or the worst case would be loss of life.

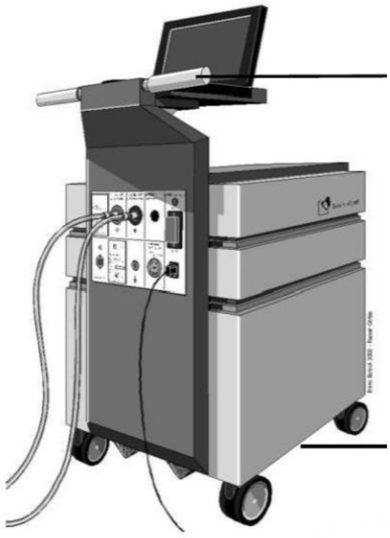


Figure 1. Cardio Workstation device (Left) and Ventricular Assist Device Model (Right) [2] [1].

Some assumptions were made, since the documentation did not fully include both the software and hardware specifications that the device has. These assumptions could be identified from the table with the (*) mark of each assumption. The first target specification which is for a Cardio Workstation model, and this specification was based on Welch Allyn's CardioPerfect Workstation [2] medical application.
Meanwhile, the second target specification, which was deployed on Hyper-V hypervirtualization, was based on Berlin Heart's EXCOR Pediatric Ventricular Assist Device [1]. Below are the tables listing the technical specifications adapted from the two devices mentioned.

	Cardio Workstation Model
Model Identifier	Desktop PC
Processor	P4
**Processor Speed*	2.7 GHz
Number of Processors	1
Hard disk	20 GB
RAM	1 GB
Ports	2 USB
Network	LAN and Wireless enabled
Operating System	Windows 10 Professional
**Version*	10.14.6
Bit	x64
**Latest Patch*	18G103

Table 6. Hardware and software specifications of the Cardio Workstation models used on VMware and VirtualBox.

	Ventricular Assist Device Model
Model Identifier	Laptop Computer
Processor	P4
**Processor Speed*	2.7 GHz
**Number of Processors*	1
**Hard disk*	20 GB
**RAM*	4 GB

Ports	1 USB
Network	LAN
Operating System	Windows 10 Professional
**Version*	10.14.6
Bit	x64
**Latest Patch*	18G103

Table 7. Hardware and software specifications of the Ventricular Assist Device model used on Hyper-V.

4.3. Hypervisors and Virtual Machines

The hypervisors tested for this project included both operating-system or hosted hypervisor as well as its counterpart; bare-metal or native hypervisor. The hosted hypervisors tested were the two most popular virtualization platforms to use, which were **VMware Workstation Pro** and **Oracle VirtualBox**. The decision to select these hypervisors came from their scope (high functionality) and cost (which both were available for free).

Next, for each of these hypervisors, three replica models of Cardio Workstation operating system were made. This is to also imitate the real-life practice of the medical device itself which is one PC running multiple VMs for multiple machines within a patient or surgery room.

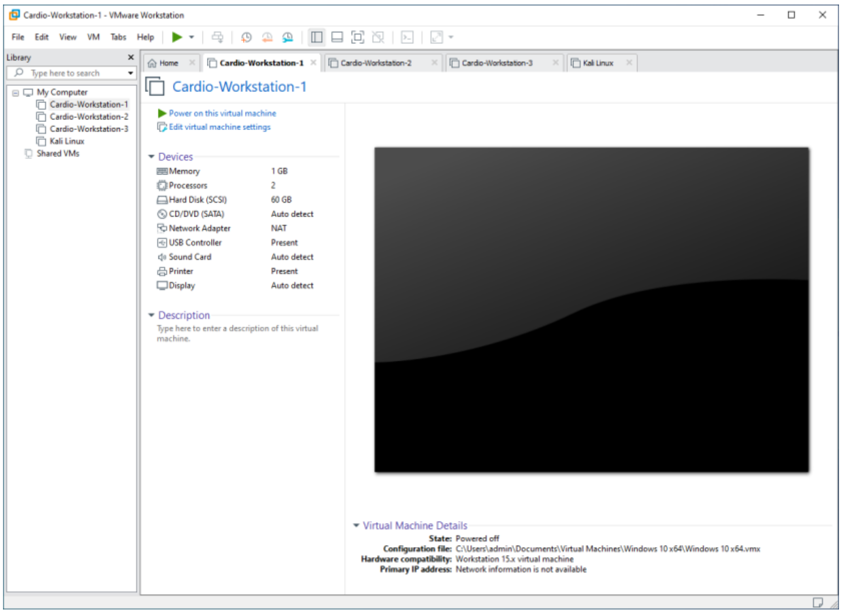


Figure 2. VMware Workstation set-up.

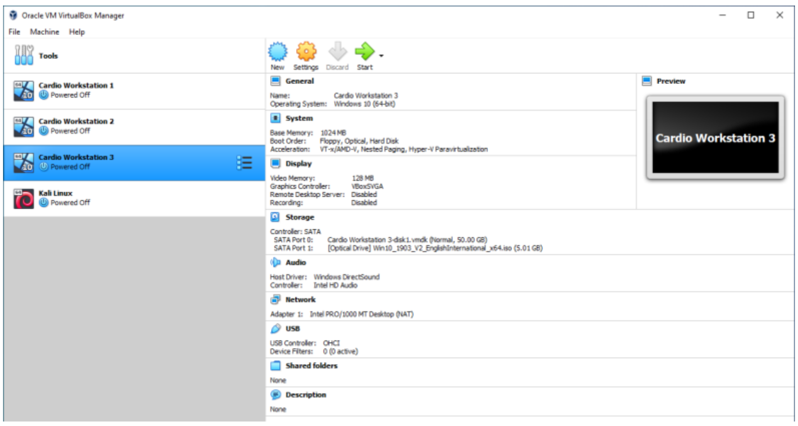


Figure 3. VirtualBox set-up.

The bare-metal or native hypervisor used was **Hyper-V** from Microsoft. The model of the Ventricular Assist Device was created on this hypervisor, and is running only one virtual machine at a time, in order to also imitate how the device works in actuality.

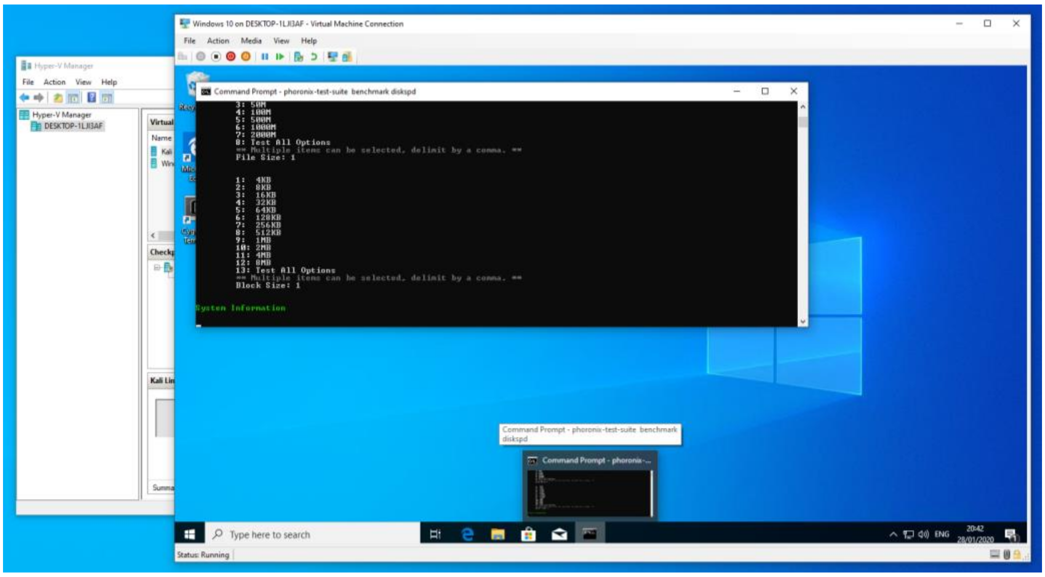


Figure 4. Hyper-V machine.

As can be seen from the previous screenshots, **Kali Linux OS** was also installed in each of the hypervisors as the penetration testing device. The reason to choose Kali Linux OS is because of the availability of vulnerability assessment and penetration testing tools in it. This made it much easier to run an automated pen-test.
To keep in mind, in the first-half of the project, all the hypervisors were using the NAT network setting. However, on the second-half of the project, a local area network design was created. This is to recreate how a malicious actor might carry out their attacks on a hospital network environment; which is typically runs on a local area network (LAN). More about this is illustrated in the figure next.

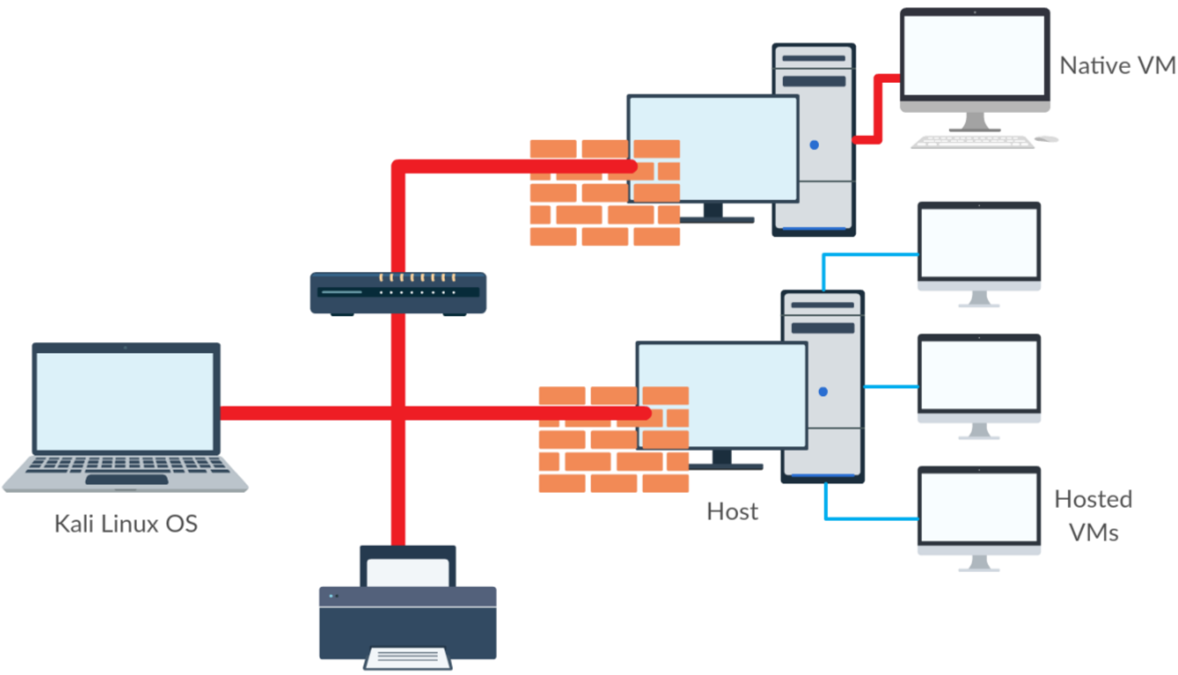


Figure 5. Simple LAN illustrated.

5. Performance Quantitative Measurements

This chapter describes the methods taken to quantitatively measure the performance for the host device, the native hypervisor machine, as well as the hosted hypervisor machines. On the table below, user can see the overview of the test-plan to quantitatively measure the host and virtual machines' performance. For further information of the tools and stress tests implemented on each system, can be read further onwards.

	Host and Virtual Machines Benchmarking
1. CPU, RAM, Disk benchmarking 2. Disk benchmarking 3. Processor benchmarking 4. System benchmarking 5. Memory benchmarking 6. CPU & Memory Stress Test (1 hour)	Run on the **Host** device and **all Virtual Machines**.
	Hypervisors Software Performance Analysis
1. Low-load Test (30 minutes) 2. Heavy-load Test (30 minutes) 3. Temperature check	Run on **all Hypervisors software**: • VMware Workstation • VirtualBox • Hyper-V

Table 8. Test plan for performance quantitative measurement.

5.1. Benchmarking

This process is done using **Novabench** [13]. The application was chosen because it has the necessary benchmark tests (CPU, disk, and memory), as well as its portability. Novabench allows USB portability use and therefore is very mobile, and very beneficial and effective to be used to regularly perform benchmark tests on medical host devices. This benchmarking test was done on the host device as well as all of the virtual machines. Novabench also supports hardware temperature monitoring and saves the temperature history, which is very useful in monitoring the fitness of the hardware.

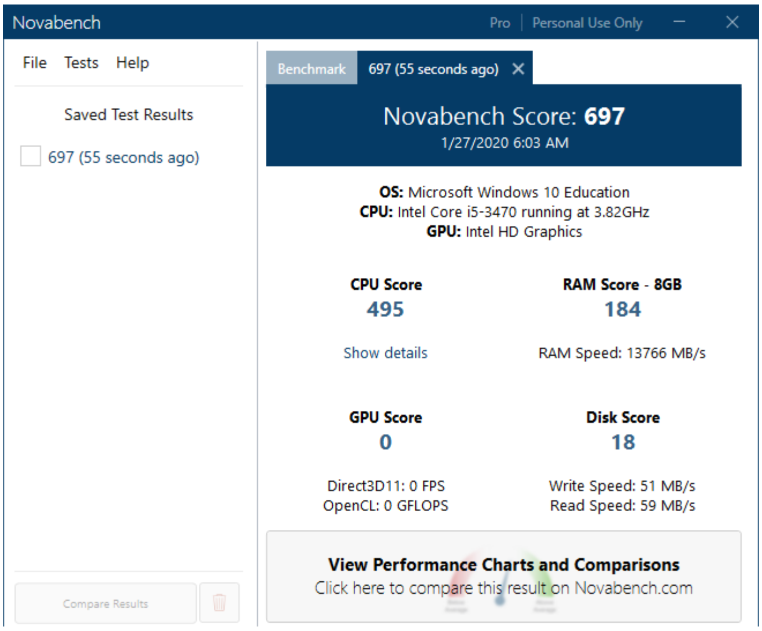


Figure 6. Novabench benchmarking application.

Additionally, the author also ran **Phoronix-Test-Suite** [14]. It is an industry standard benchmarking and stress test library that is available for free, created by Phoronix Media that also supports OpenBenchmarking.org: the biggest open and collaborative testing platform. The author selected three tests, which can be seen from the table below [15].

windows/diskspd	Windows OS disk benchmarking
windows/openssl	Windows OS Open SSL processor benchmarking
windows/redis	Windows OS Redis system benchmarking
pts/t-test1	T-Test memory benchmarking

Table 9. Stress tests parameters.

The reason of choosing Phoronix-Test-Suite as a benchmark tool, is because it is fully customizable. And since the main concerns of a medical device would be processor, disk, system, and memory capabilities, then it will not make sense to use other benchmark tools which were marketed towards other uses (e.g. for GPU performance benchmarking). Below is a screenshot of Phoronix-Test-Suite in action.

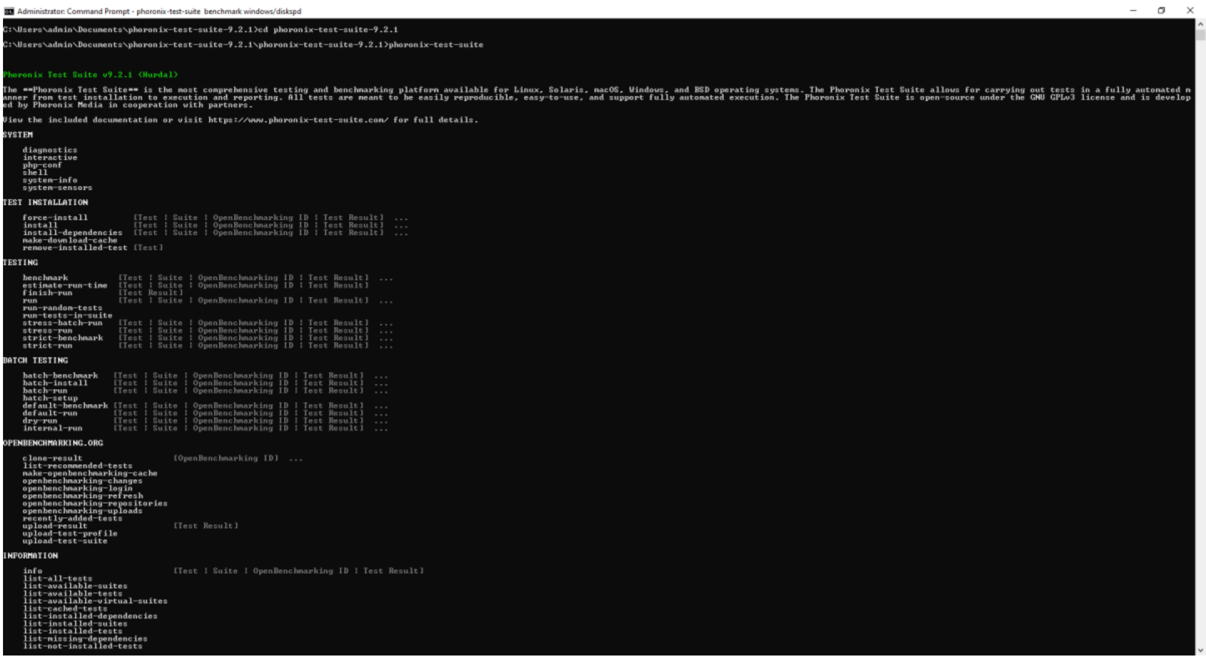


Figure 7. Phoronix-Test-Suite command line interface.

5.2. Performance Analysis

The tool used to perform in-depth analysis of the performance for each hypervisors is Intel **VT Profiler 2020** [16]. This tool is also available for free, and performs a detailed profiling and monitoring of the hypervisors (as well as other executable applications).

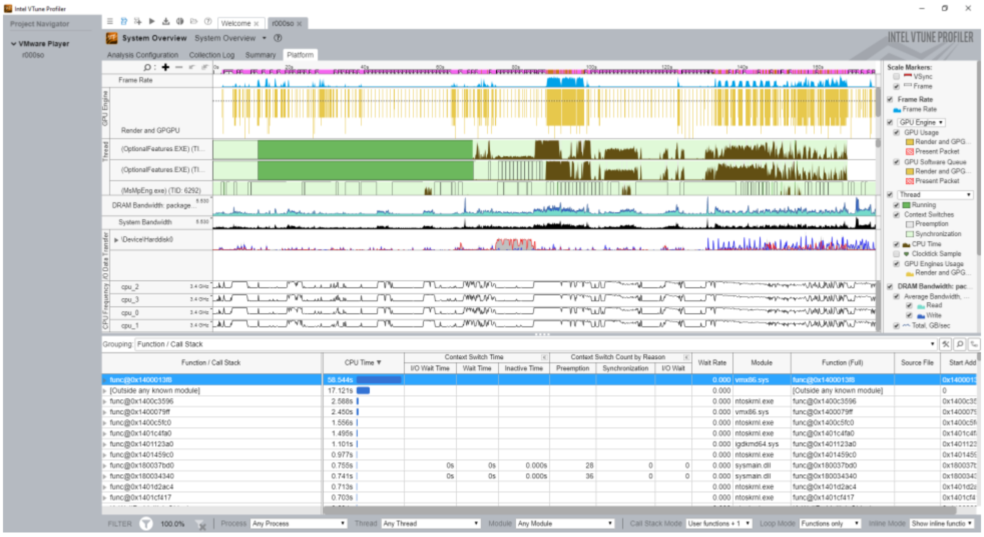


Figure 8. Intel VT Profiler 2020 to analyse deeply the hypervisors software.

Using the Intel Profiler, there were two experiments done for each hypervisors. This can be viewed at the following table.

Low-load Test	**Heavy-load Test**
• 1 VM running • Run CPU stress test (30 minutes) • Run memory stress test (30 minutes)	• 3 VMs running symultaneously • Run CPU stress test on each VM (30 minutes) • Run memory stress test on each VM (30 minutes)

Table 10. Performance analysis test specification.

The stress test tool used is **HeavyLoad** [17]. The tool was selected due to its GUI that makes it easier to monitor the running stress tests. This tool is also available for free and for commercial use. The step by step of this test-plan can be watched on Youtube, via this link:

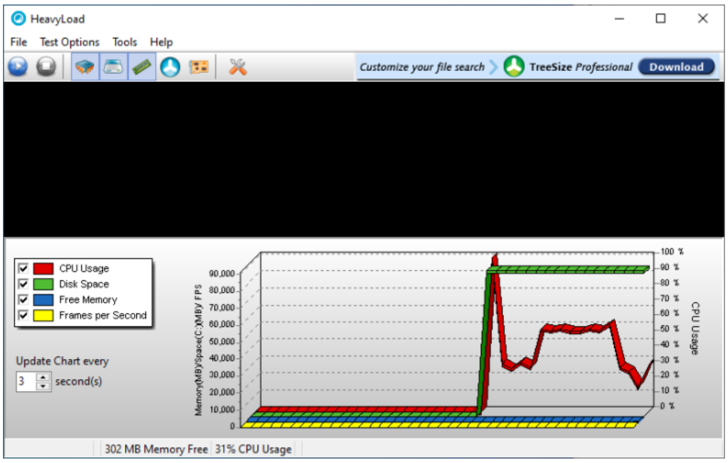


Figure 9. HeavyLoad application.

[https://www.youtube.com/playlist?list=PLVITNI4tfD9XDU0NPfJULE9ZjWRrmEG11.](https://www.youtube.com/playlist?list=PLVITNI4tfD9XDU0NPfJULE9ZjWRrmEG11)

6. Security Assessment

This chapter is comprised of two parts, and they are vulnerability scanning and penetration testing. This part of the project is to make sure that the proposed hypervisor solution for medical devices applications is secure and reliable.

6.1. Vulnerability Scanning

The vulnerability scanning was done using **Nessus Essentials** [18], which is available for free from Tennable. Nessus Essentials, however, is limited to be used for 16 IPs under one registered account. Other than this, it is a high quality software to use to discover vulnerabilities both for the host device as well as the virtual machines.

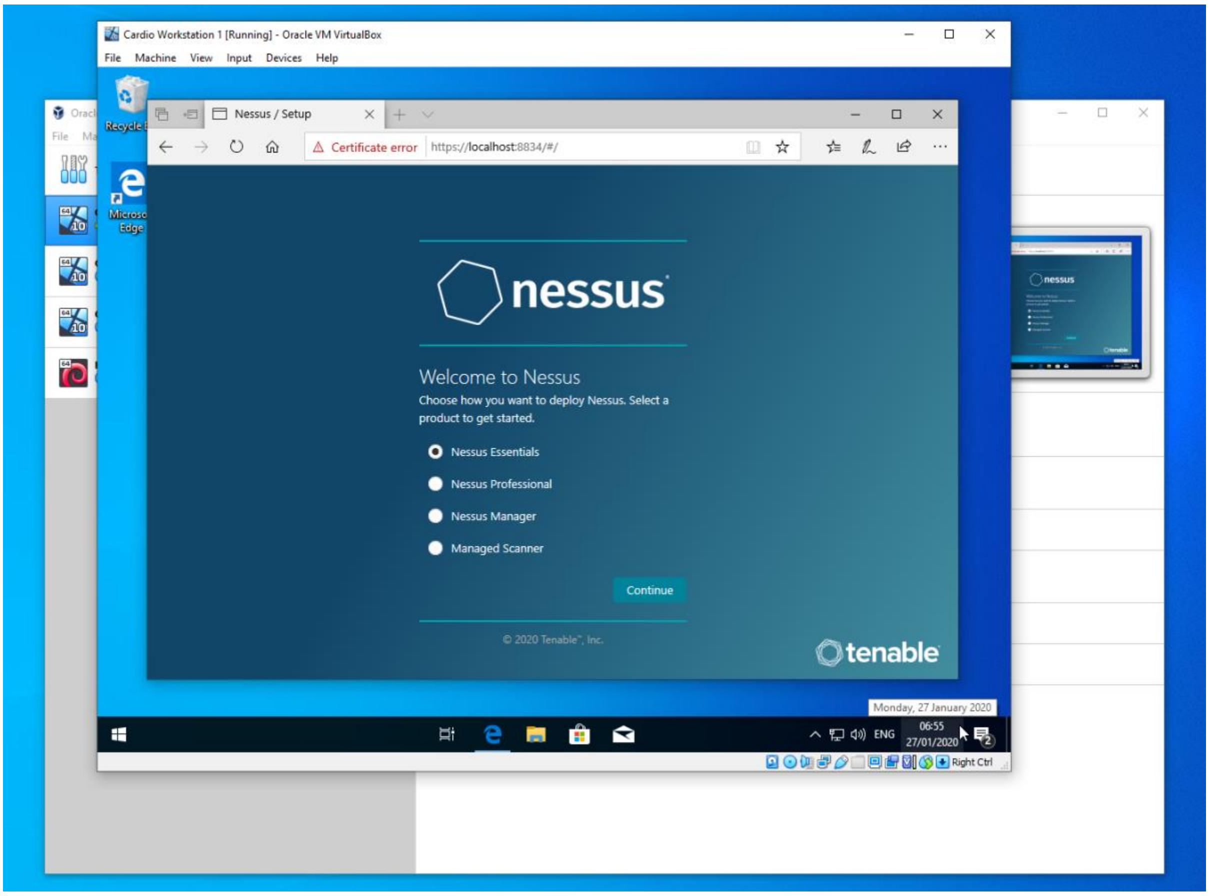


Figure 10. Nessus Essential for vulnerability scanning.

6.2. Penetration Testing

During the penetration testing, all virtual machines (both from the hosted and native hypervisors) were tested in addition to the host device. This is to make sure that the device (as medical devices) can stay up and available even during attacks, which is a highly critical requirement for assistive medical devices.

The penetration testing itself consists of network scanning on frequently used ports, aggressive vulnerability scanning, as well as fuzzing. All of the scripts used were written, customized, and set-up by the author of this project.

Meanwhile, the tools used in assistance were all open licensed and already available within **Kali Linux OS** [19]. For the network scanning and vulnerability scanning, the author used **Nmap** [20] and **Metasploit** [21] Framework. As for the **fuzzing,** the script was written in **Python 3** and is using **Boo-Fuzz framework** [22]. The full description of the pen-test plan can be seen at the table below.

TCP Scans

Gather information passively using non-authorized user role, and assess possible ICS (Internet Connection Sharing) vulnerabilities.
<u>TCP ports scanned</u>:
7, 13, 17, 19, 20, 21, 23, 25, 42, 53, 80, 88, 102, 110, 119, 135, 139, 389, 443, 445, 464, 515, 548, 554, 563, 593, 636, 647, 1024-5000, 6600, 8080, 9389, 42424, 49152-65535.

UDP Scans

Scan general UDP ports, and
assess possible ICS (Internet Connection Sharing) vulnerabilities on low-medium UDP ports.
<u>UDP ports scanned:</u>
7,9,13,17,19,20-25, 42,53,67,69,80, 88, 123, 137, 138, 139, 161, 162, 389, 443-445, 464, 500, 502, 530, 593, 789, 1024-5000, 5004, 8080, 9600, 19999, 20000, 20547, 34962-34964, 34980, 44818, 46823, 46824, 49152-65535.

OS Scans

Check if device has MS12-008 and MS10-073 vulnerabilities,
gather information actively using non-authorized user-role,
retrieve NetBIOS name from device.

SMB Scans

List supported protocols of device's SMB server,
check if device is vulnerable to MS10-096, MS10-074,
MS16-075, CVE-2017-8543 & CVE-2017-8589,
and check SMB message signing configuration.

SMB Enumerations

Enumerate Windows user accounts using SAMR,
enumerate Windows user accounts using LSA,
enumerate SMB sessions,
and enumerate SMB shares.

Bruteforce

Attempt bruteforce attack through SSH client,
bruteforce attack shared folders using anonymous credentials,
bruteforce attack shared folders using authorized credentials,
bruteforce shared users using SMB brute script,
bruteforce shares using SMB brute script,
bruteforce SMB login protocol,
check possible remote login and execution.

Launch Payloads and Active Pen-testing

Set up windows/shell_reverse_tcp payload,
set up windows/x64/meterpreter_reverse_tcp payload to use for exploits,
set up generic/shell_reverse_tcp payload to use for exploits,
attempt to login to SMB by using psexec utility,
check vulnerability MS15-034 and SMB vulnerability MS10-054,
perform Denial of Service on SMB vulnerability MS10-054,
check SMB vulnerability MS10-061,
try to exploit SMB vulnerability MS10-061,
check SMB vulnerability MS17-010,
scan SMB vulnerability MS17-010 through auxiliary,
exploit SMB vulnerability MS17-010 through auxiliary,
try exploiting possible SMB vulnerability MS17-010,
check RDP vulnerability MS-12-020,
scan RDP vulnerability MS12-020 through auxiliary,
exploit possible RDP vulnerability MS12-020,
perform SYN Flood DOS attack,
start payload handler windows/meterpreter/reverse_tcp to start a session,
start payload handler windows/shell_reverse_tcp to start a session,
start payload handler windows/x64/meterpreter_reverse_tcp
to start a session,
start payload handler generic/shell_reverse_tcp to start a session,
list any active sessions opened by the handlers,
check if device has MS13-081 vulnerability,

perform Blue Screen of Death (BSOD) exploit, *check if device has MS15-004 vulnerability by performing patches enumeration,* *check if device has MS15-004 vulnerability by performing remote desktop service attack,* *check if device has MS14-060 Sandworm vulnerability,* *and kill running jobs.*
Fuzzing
Simple fuzzing with “user”, “pass”, “stor”, and “retr” arguments.

Table 11. Automated pen-test plan.

This penetration testing tool can fully be replicated on other Kali Linux OS (and other Linux OS), the installation instruction with the scripts can be downloaded from: https://github.com/fauziahadhim/OSSEC-MedDevice-PenTest.

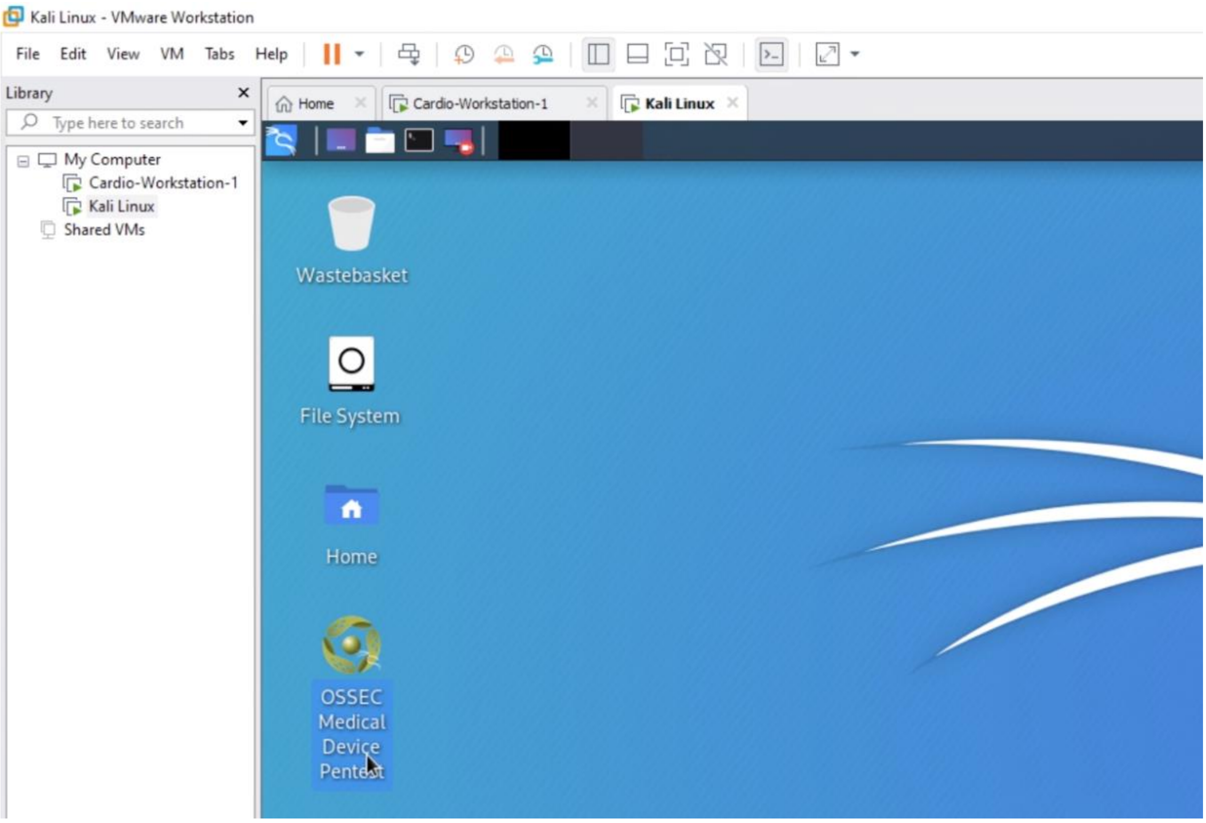


Figure 11. Automated pen-test tool for the project.

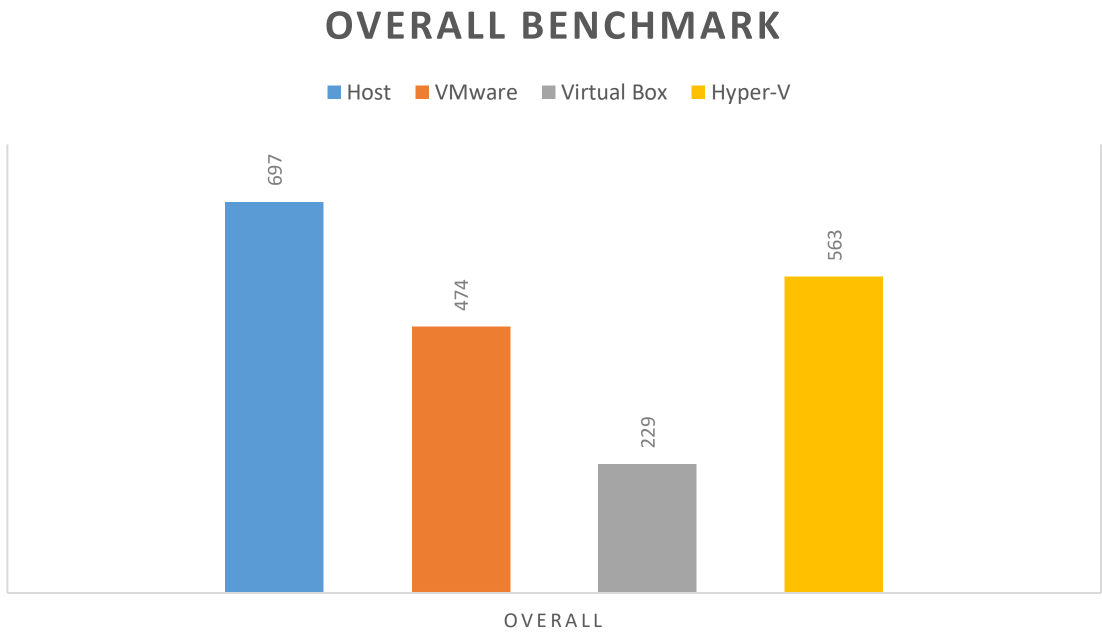
The tool automatically will save into .txt files, for the most recent test results in /root/Documents/OSSEC-MedDevice-PenTest directory according to the type of test carried out.

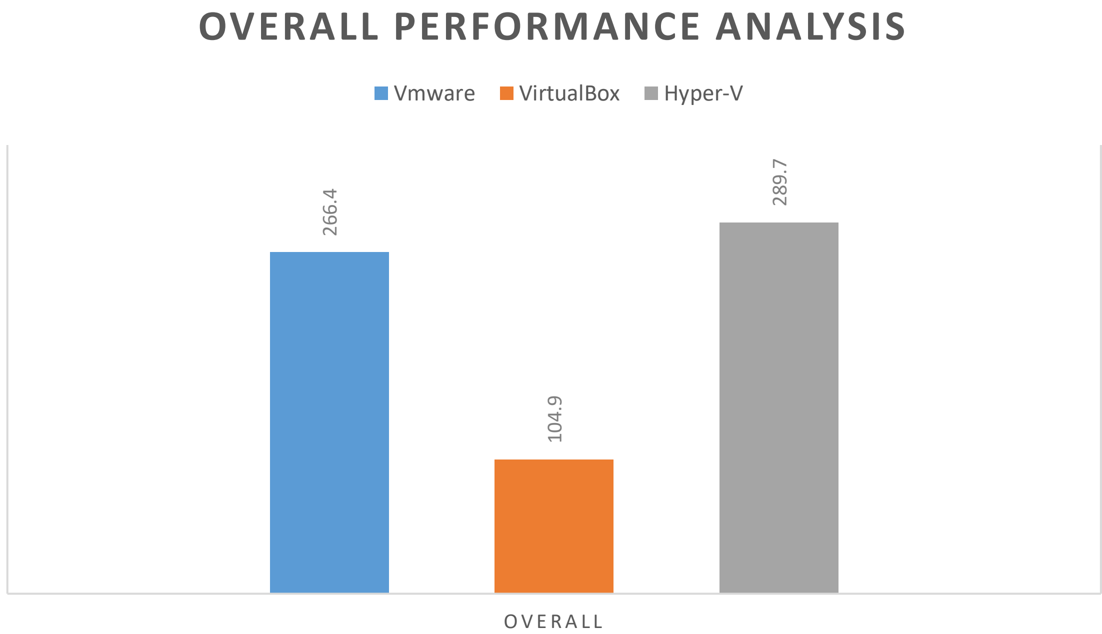
7. Results

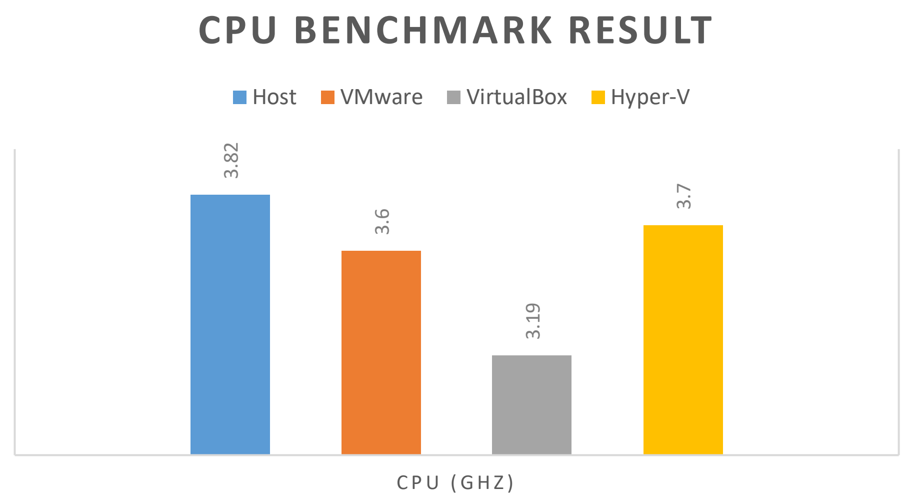
The results are divided into two sections, each for the performance quantitative measurements and the security analysis result.

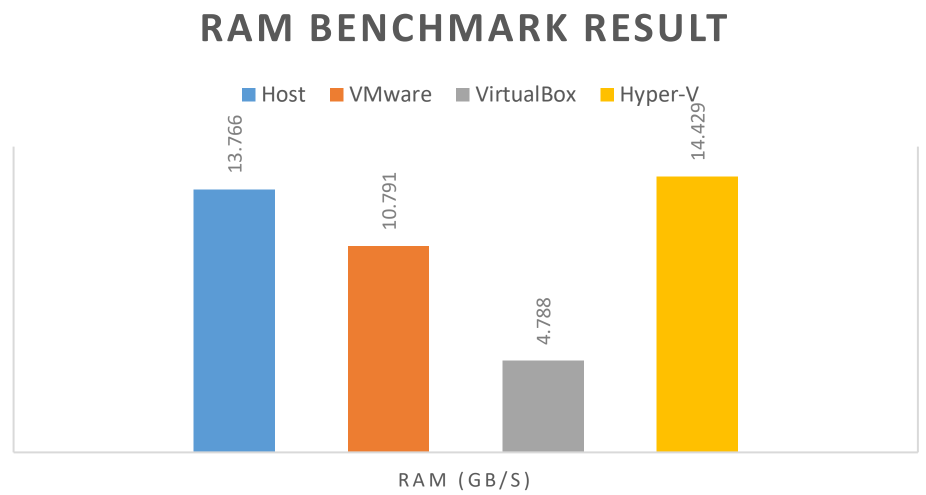
7.1. Performance Quantitative Measurements

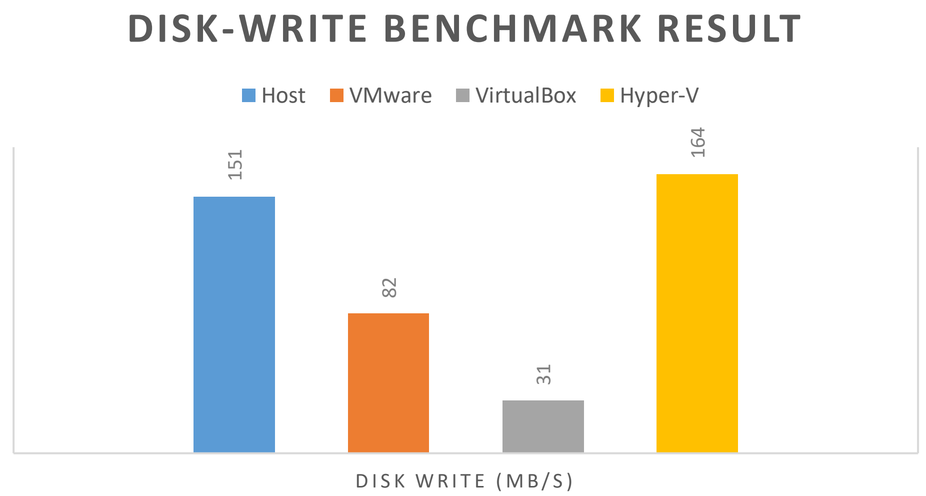
In addition to the screenshots of the tools' interpretations, comparisons are done with simplified diagrams, to make it easier to digest and create conclusion from the experiments.

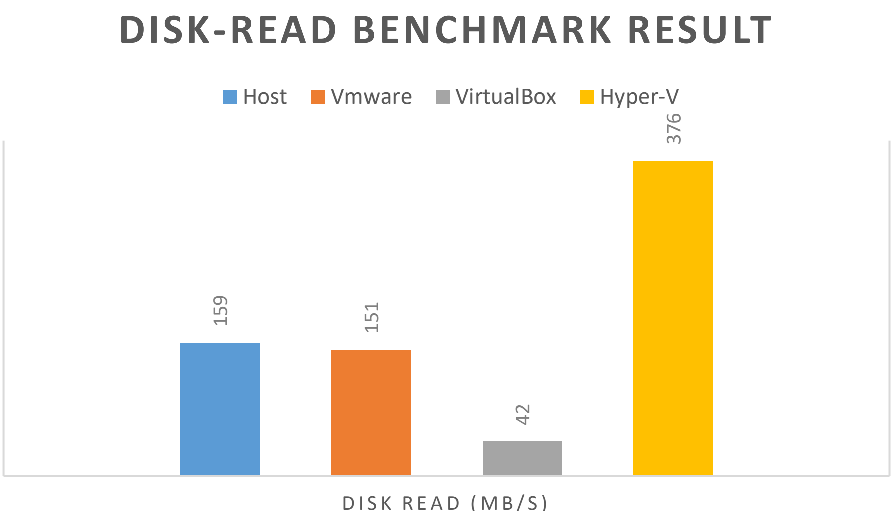


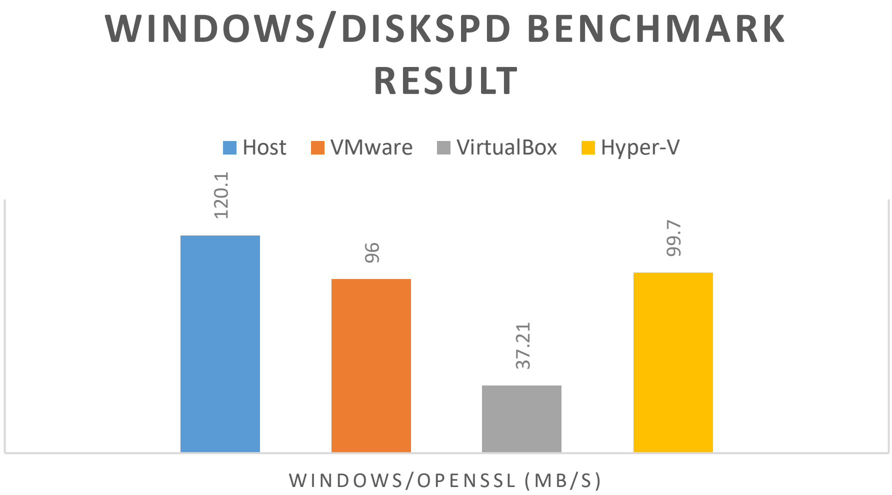


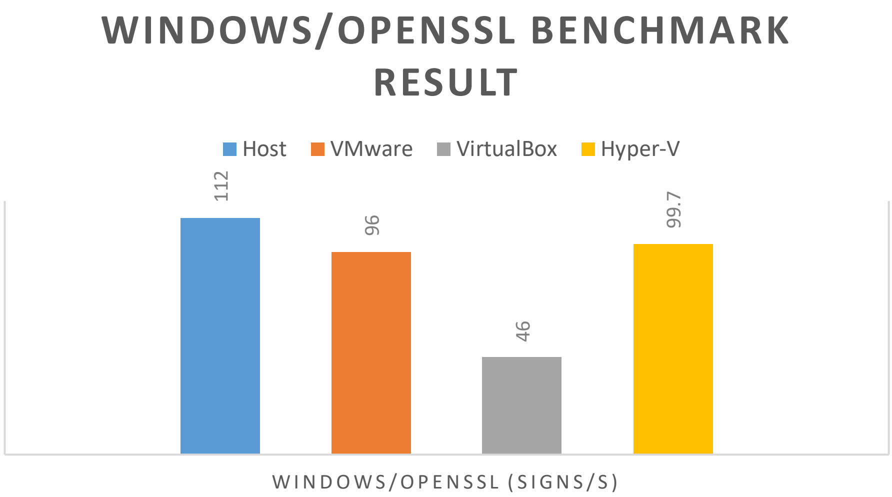

CPU BENCHMARK RESULT
Host
VMware
VirtualBox
Hyper-V
3.82
3.6
3.19
3.7
CPU (GHZ)

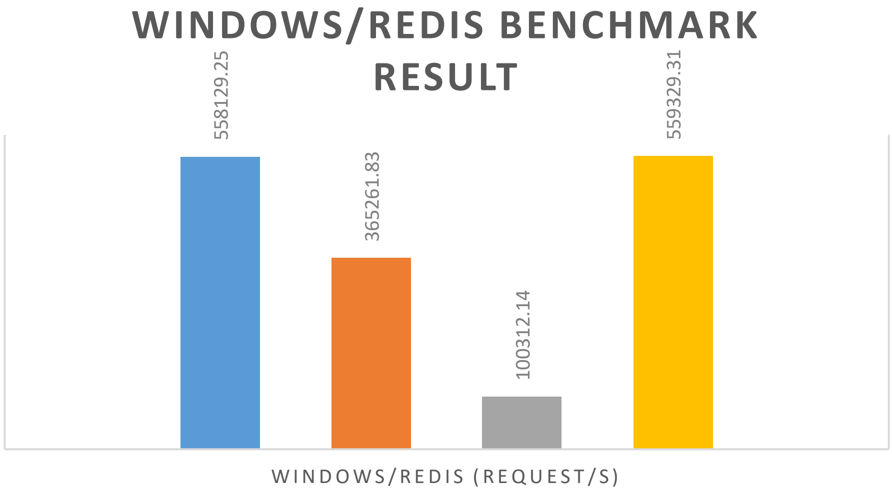

RAM BENCHMARK RESULT
Host
VMware
VirtualBox
Hyper-V
13.766
10.791
4.788
14.429
RAM (GB/S)

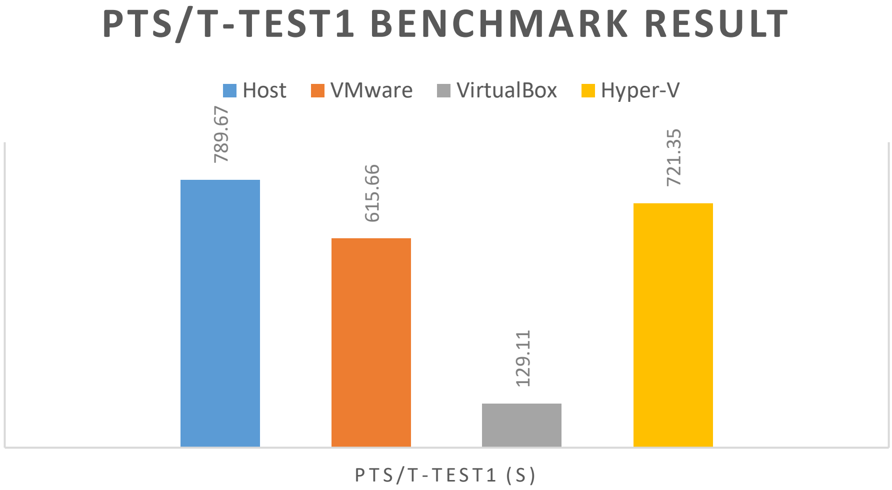

DISK-WRITE BENCHMARK RESULT
Host
VMware
VirtualBox
Hyper-V
151
82
31
164
DISK WRITE (MB/S)

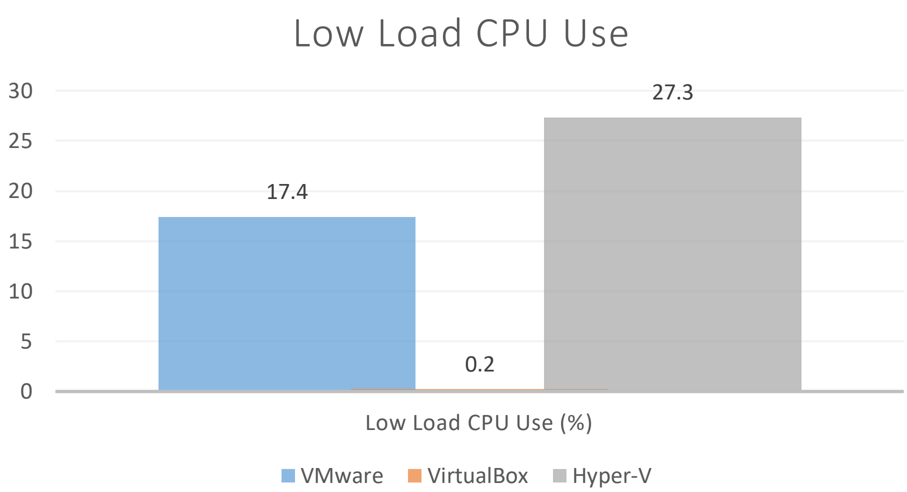

DISK-READ BENCHMARK RESULT
Host
Vmware
VirtualBox
Hyper-V
159
151
42
376
DISK READ (MB/S)

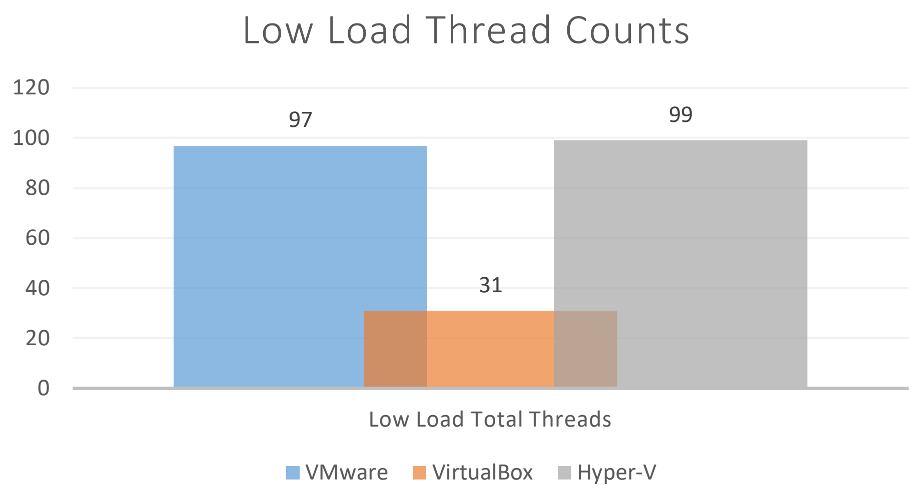

WINDOWS/DISKSPD BENCHMARK RESULT
Host
VMware
VirtualBox
Hyper-V
120.1
96
37.21
99.7
WINDOWS/OPENSSL (MB/S)

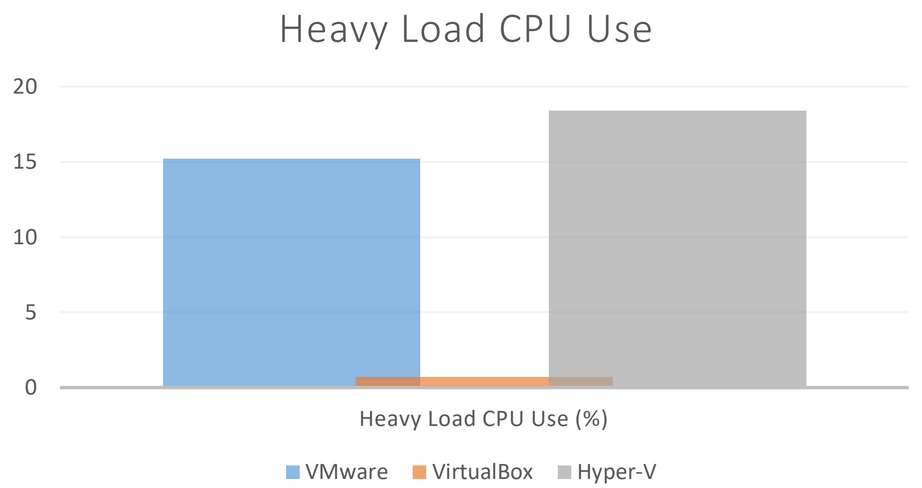

WINDOWS/OPENSSL BENCHMARK RESULT
Host
VMware
VirtualBox
Hyper-V
112
96
46
99.7
WINDOWS/OPENSSL (SIGNS/S)

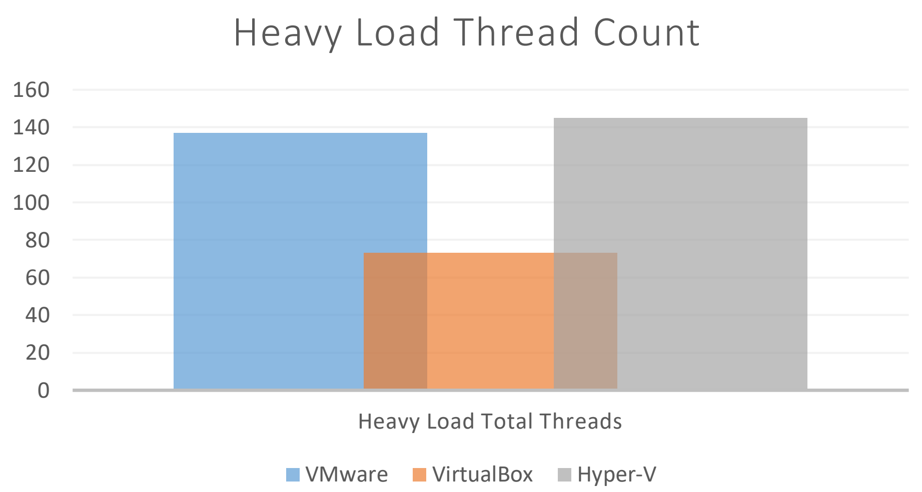

WINDOWS/REDIS BENCHMARK RESULT
558129.25
365261.83
100312.14
559329.31
WINDOWS/REDIS (REQUEST/S)


PTS/T-TEST1 BENCHMARK RESULT
Host
VMware
VirtualBox
Hyper-V
789.67
615.66
129.11
721.35
PTS/T-TEST1 (S)


Low Load CPU Use
30
25
20
15
10
5
0
17.4
0.2
27.3
Low Load CPU Use (%)
VMware
VirtualBox
Hyper-V







7.2. Security Analysis

The results from Vulnerability Analysis using Nessus Essential, as well as using the automated Pen-test tool, found that the system has vulnerable SMB setting. Which is quiet common in devices that need to connect to a Local Area Network (LAN).

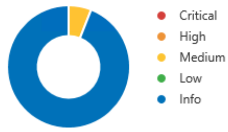
Below is the recap of the vulnerabilities found.

Sev	Name	Family	Count
MEDIUM	SMB Signing not required	Misc.	1
INFO	DCE Services Enumeration	Windows	8
INFO	4 SMB (Multiple Issues)	Windows	5
INFO	Authenticated Check : OS Name and Installed Package Enu...	Settings	1
INFO	Common Platform Enumeration (CPE)	General	1
INFO	Device Type	General	1
INFO	Host Fully Qualified Domain Name (FQDN) Resolution	General	1
INFO	Local Checks Not Enabled (info)	Settings	1
INFO	Microsoft Windows NTLMSSP Authentication Request Rem...	Windows	1
INFO	Nessus Scan Information	Settings	1
INFO	No Credentials Provided	Settings	1
INFO	OS Identification	General	1
INFO	OS Identification and Installed Software Enumeration over ...	Misc.	1

Scan Details

Policy:	Basic Network Scan
Status:	Completed
Scanner:	Local Scanner
Start:	Today at 11:23 AM
End:	Today at 11:34 AM
Elapsed:	10 minutes

Vulnerabilities



There were some open ports detected as well, however, they are filtered, and will require further analysis of the software/application using it, in order to carry an attack.

```
PORT      STATE         SERVICE         REASON
20/udp    open|filtered ftp-data        no-response
21/udp    open|filtered ftp             no-response
22/udp    open|filtered ssh             no-response
23/udp    open|filtered telnet          no-response
24/udp    open|filtered priv-mail       no-response
25/udp    open|filtered smtp            no-response
80/udp    open|filtered http            no-response
102/udp   open|filtered iso-tsap        no-response
137/udp   open          netbios-ns      udp-response ttl 128
138/udp   open|filtered netbios-dgm     no-response
139/udp   open|filtered netbios-ssn     no-response
443/udp   open|filtered https           no-response
444/udp   open|filtered snpp            no-response
445/udp   open|filtered microsoft-ds    no-response
502/udp   open|filtered mbap            no-response
530/udp   open|filtered courier         no-response
593/udp   open|filtered http-rpc-epmap  no-response
789/udp   open|filtered unknown         no-response
8080/udp  open|filtered http-alt        no-response
9600/udp  open|filtered micromuse-ncpw  no-response
19999/udp open|filtered dnp-sec         no-response
20000/udp open|filtered dnp             no-response
20547/udp open|filtered unknown         no-response
34980/udp open|filtered ethercat        no-response
44818/udp open|filtered EtherNetIP-2    no-response
46823/udp open|filtered unknown         no-response
46824/udp open|filtered unknown         no-response
MAC Address: 00:0C:29:0B:C8:6F (VMware)
```

Figure 12. List of open ports found.

Fuzzing was done using the character constructions mentioned in earlier chapter. The VMware and the Hyper-V stayed up, and no failure case were found.

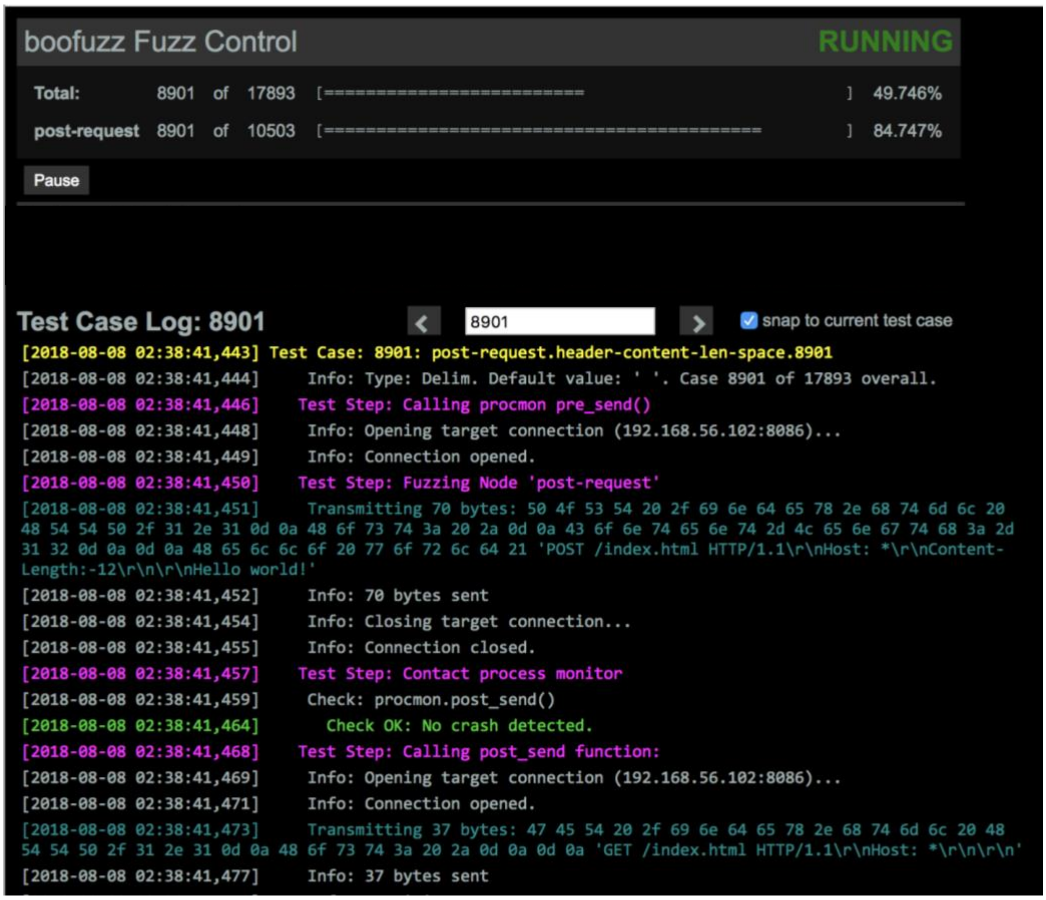


Figure 13. Fuzzer in action from the web browser interface.

However, unfortunately for the Virtual Box, after the test ran for about 1.5 hour, the VirtualBox machine shut down suddenly, and the famous blue screen appeared.

Due to this unfortunate event, the data for the automated pen-test result for VirtualBox was corrupted. And to run another try, the VM was too heavy and often freezed. Therefore the author decided that the VirtualBox was a lost cause and were not able to proceed with the pen-test.

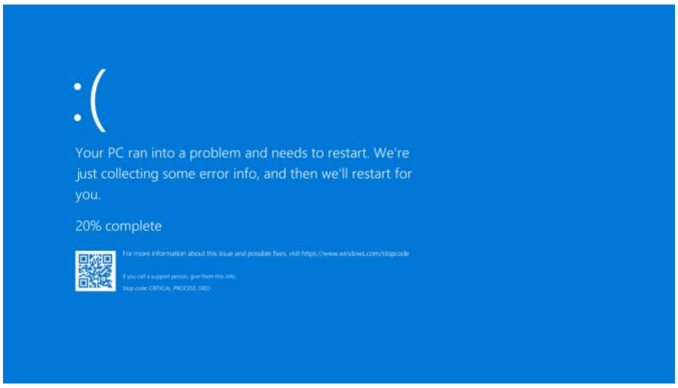


Figure 14. The famous blue screen.

8. Recovery Plan

A recovery plan checklist has been made according to the found vulnerabilities, which can be applied when an emergency situation happened, especially for network or access hijack cases. The recovery plan is very easy to follow and are referenced from official Microsoft developer website. The check list can be viewed below, and of course it is fully up to the users/admin for which steps they take from the available options.

Recovery Plan – Windows 10 1903

☐ **Disable SMBv1.**

Precondition:
- Given ‘administrator’ privilege.
- Default configurated (SMBv1 is enabled)

Set-up: PowerShell 2.0 or later.

```
Get-Item
HKLM:\SYSTEM\CurrentControlSet\Services\LanmanServer\Paramete
rs | ForEach-Object {Get-ItemProperty $_.pspath}

Set-itemProperty -Path
"HKLM:\SYSTEM\CurrentControlSet\Services\LanmanServer\Paramet
ers" SMB1 -Type DWORD -Value 0 -Force
```

Reference:
https://support.microsoft.com/en-us/help/2696547/detect-enable-disable-smbv1-smbv2-smbv3-in-windows-and-windows-server

☐ **Block the following ports from the Internet:**
- ☐ **23/tcp**
- ☐ **139/tcp**
- ☐ **445/tcp**

Precondition:
- Given ‘administrator’ privilege.

Set-up: Control Panel, Windows Firewall.

```
Start > Control Panel > Windows Firewall > Advanced settings
> Advanced Security > Inbound Rules > New Rule > Select
mentioned ports
```

☐ **Stop print spooler service.**

Precondition:

- Given ‘administrator’ or ‘user’ privilege.

Set-up: Task Manager.

```
Task Manager > Services > Spooler (Print Spooler) > Stop
```

☐ **Block the following ports from the Internet:**
  ☐ **515/tcp**

Precondition:
- Given ‘administrator’ privilege.

Set-up: Control Panel.

```
Start > Control Panel > Windows Firewall > Advanced settings
> Advanced Security > Inbound Rules > New Rule > Select
mentioned ports
```

☐ **Disable Windows Installer Service.**

Precondition:
- Given ‘administrator’ or ‘user’ privilege.

Set-up: Task Manager.

```
Task Manager > Services > Open Services >
Services (Local) > Windows Installer > Stop
```

☐ **Disable ‘Guest’ account.**

Precondition:
- Given ‘administrator’ privilege.

Set-up: Local User Manager.

```
Start > Edit local users and groups >
Users > Guest > Properties > Account is disabled
```

☐ **Filter LoadLibrary from third parties and remote shares.**

Precondition:
- Given 'administrator' privilege.

Set-up: Process Monitor. https://docs.microsoft.com/en-us/sysinternals/downloads/procmon

```
Select specific directories and file handler assigned to the
application/device.
```

References:https://support.microsoft.com/en-us/help/2389418/secure-loading-of-libraries-to-prevent-dll-preloading-attacks

---

☐ **Set Rules for Outbound in Windows firewall:**
  ☐ **Block possible malicious sites.**

Precondition:
- Windows 10 is used.
- Given 'administrator' privilege.

Set-up: Windows Defender.

```
Start > Windows Defender > Firewall & network protection >
Advanced setting > Inbound Rules > New rule > Custom > Next >
Program > All programs > Protocol and Ports > Select protocol
and port number >
Scope > Next > Action > Disable connection > Next >
Profile > Select network location type: "Public" > Next >
Name > Type name and description of rule > Finish
```

References:
https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-firewall/create-an-inbound-port-rule

---

☐ **Block 3389/tcp from the Internet.**

Precondition:
- Given 'administrator' privilege.

Set-up: Control Panel.

```
Start > Control Panel > Windows Firewall > Advanced settings
> Advanced Security > Inbound Rules > New Rule > Select
mentioned ports
```

☐ **Allow only selected remote computers to be connected to Remote Desktop Protocol.**

Precondition:
- Given ‘administrator’ privilege.

Set-up: Control Panel.

```
Start > Control Panel > Administrative Tools > Local Security
Policy >
Local Policies > User Rights Assignment > Allow logon through
Remote Desktop Services > Allow only desired (e.g.
administrator)
```

☐ **Set account lookout policy**

Precondition:
- Given ‘administrator’ privilege.

Set-up: Control Panel.

```
Start > Control Panel > Administrative Tools > Local Security
Policy >
Account lookout policies > Set "3 attempts with 3 minute
lockout durations" values for all 3 options
```

☐ **Set a password for user account**

Precondition:
- Given ‘administrator’ privilege.

Set-up: Control Panel.

```
Start > Control Panel > User Accounts >
User Accounts > Change your Windows Password >
Create a password for your account
```

References:
https://support.microsoft.com/en-us/help/17463/windows-7-connect-to-another-computer-remote-desktop-connection

☐ Enable Open File Security warning.

Precondition:
- Given 'administrator' privilege.

Set-up: Control Panel.

```
Start > inetcpl.cpl > Security > Security level for this zone
> Custom level … >
Security Setting > Launching application and unsafe files
(not recommended) > Prompt
```

☐ Disable Internet Explorer.

Precondition:
- Given 'administrator' privilege.

Set-up: Run.

```
Start > Run > appwiz.cpl > OK >
Programs and Features > Turn Window features on or off >
Windows Feature > Internet Explorer > Untick the check box >
OK
```

References:
https://support.microsoft.com/en-us/help/4013567/how-to-disable-internet-explorer-on-windows

☐ Turn on SmartScreen filter.

Precondition:
- Given 'administrator' privilege.

Set-up: Microsoft Edge.

```
Microsoft Edge > Settings > Advanced settings > View advanced
settings >
Help protect me from malicious sites and downloads with
Windows Defender SmartScreen > On
```

☐ **Stop Web Proxy Auto Discovery Protocol (WPAD).**

Precondition:

- Given ‘administrator’ privilege.

Set-up: Control Panel.

```
Start > Control Panel > Network and Internet > Internet
Options > Internet Properties > Connections > LAN Settings >
Untick Automatically detect settings > OK
```

9. Conclusion

From the performance quantitative measurement, we can conclude it with the bar charts below. Overall, Hyper-V and VMware won. This can be an option if companies want to implement native or hosted hypervisor on their medical devices. Of course, the best solution is offered by Hyper-V as a native hypervisor.

As for creating a secure environment, a scheduled vulnerability scanning as well as an automated pen-test should be carried out regularly. Possibly this should be done at least once every 6 months, to make sure that the devices are always monitored and recorded for their patches/updates.

Further implementation that can be implemented, would be in a form of a secure private VPN connection by implementing DMZ on hospital sections with high-risk or in critical condition patients. Of course, this would take a lot more time to create such network design. One of the methods that may help to bring this idea to practice, is using Cisco Packet Tracer. Perhaps, in future projects this can be a topic to explore.

Some limitations encountered during the making of this project, is to use tools which can be used hand-in-hand with a native hypervisor. Many open-licensed or freely available tools are not supporting this Hyper-V service. An example is the Intel VTune Profiler. Thankfully, Microsoft provided a solution towards this [23]. However, it would be nice to use the time spent for tinkering this, to maybe used for expanding the project scope.

I learned a lot for the architecture of operating system and how the processor handles tasks. And during the performance analysis, it was something very new to learn about multi-threading more specifically on Hypervisors. I have also never done benchmarking nor experimenting with hypervisors, so these are very insightful and I know I will use this a lot in the future.

Overall, I am quite satisfied with the project. I learned a lot of new topics and methods. Also, I was able to use the knowledge I already had before which was on penetration testing. As I mentioned earlier, there are some more things I wished to implement, however, this will be for the future.

10. Bibliography

[1] Berlin Heart GmbH, “EXCOR® Pediatric VAD Ventricular Assist Device with Stationary Driving Unit Ikus Rev. 2.1 Instructions for Use 1000721x09 Revision 8,” 2020. [Online]. Available: https://www.berlinheart.de/en/medical-professionals/excorr-pediatric/. [Accessed 18 January 2020].

[2] Welch Allyn, “PC-Based Resting Electrocardiograph,” [Online]. Available: https://www.welchallyn.com/en/products/categories/cardiopulmonary/resting-ecg/cardioperfect-pc-based-resting-electrocardiograph.html. [Accessed 18 January 2020].

[3] Welch Allyn, Inc., “CP 200™ 12-Lead Resting Electrocardiograph,” Welch Allyn, Skaneateles Falls, 2008.

[4] Abbott, “CentriMag Acute Circulatory Support System,” 2020. [Online]. Available: https://www.cardiovascular.abbott/us/en/hcp/products/heart-failure/centrimag-acute-circulatory-support-system/ht-tab/tech.html. [Accessed 21 January 2020].

[5] Carl Zeiss Meditec AG, “Zeiss IOLMaster 700,” 2020. [Online]. Available: https://www.zeiss.com/meditec/int/product-portfolio/optical-biometers/iolmaster-700.html?pos=productfinder_sites_wordcloud_iol-master#downloads. [Accessed 21 January 2020].

[6] U.S. Food & Drug Administration, “Abbott Recalls CentriMag Circulatory Support System Motor Due to Pump and Motor Issues,” 6 November 2019. [Online]. Available: https://www.fda.gov/medical-devices/medical-device-recalls/abbott-recalls-centrimag-circulatory-support-system-motor-due-pump-and-motor-issues.

[7] Dicardiology, “Abbott Recalls CentriMag Circulatory Support System After 44 Injuries and One Death,” 4 November 2019. [Online]. Available: https://www.dicardiology.com/content/abbott-recalls-centrimag-circulatory-support-system-after-44-injuries-and-one-death.

[8] D. Zhou and Y. Tamir, “Fast Hypervisor Recovery Without Reboot,” in *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Luxembourg City, 2018.

[9] DataForth, “Protecting Signal Lines Against Electromagnetic Interference,” [Online]. Available: https://www.dataforth.com/protecting-signal-lines-against-electromagnetic-interference.aspx.

[10] N. Davé, “Cyberattacks on Medical Devices Are on the Rise—and Manufacturers Must Respond,” 17 December 2019. [Online]. Available: https://spectrum.ieee.org/the-human-os/biomedical/devices/cyber-attacks-on-medical-devices-are-on-the-riseand-manufacturers-must-respond.

[11] B. L. Jr., “FDA issues warning on medical devices that are vulnerable to takeover from hackers,” 1 October 2019. [Online]. Available: https://www.cnbc.com/2019/10/01/fda-issues-warning-on-medical-devices-that-are-vulnerable-to-cyberattacks.html.

[12] European Commission, “ec.europa.eu,” [Online]. Available: https://ec.europa.eu/growth/sectors/medical-devices/new-regulations_en. [Accessed 17 January 2020].

[13] Novawave Inc., “Novabench,” 2020. [Online]. Available: https://novabench.com/. [Accessed 21 January 2020].

[14] Phoronix Media, “Phoronix Test Suite,” 2020. [Online]. Available: https://www.phoronix-test-suite.com/. [Accessed 20 January 2020].

[15] Phoronix Media, “Phoronix Test Suite Documentation,” 2020. [Online]. Available: https://www.phoronix-test-suite.com/documentation/phoronix-test-suite-windows.html. [Accessed 20 January 2020].

[16] Intel Corporation, “software.intel.com,” 2020. [Online]. Available: https://software.intel.com/en-us/vtune. [Accessed 21 January 2020].

[17] JAM Software GmbH, “jam-software.com,” 2020. [Online]. Available: https://www.jam-software.com/heavyload. [Accessed 21 January 2020].

[18] Tenable, Inc., “The Nessus Family,” 2020. [Online]. Available: https://www.tenable.com/products/nessus. [Accessed 21 January 2020].

[19] Offensive Security, “Kali Linux Downloads,” 2020. [Online]. Available: https://www.kali.org/downloads/. [Accessed 19 January 2020].

[20] Nmap Security Scanner, “Nmap,” 2020. [Online]. Available: https://nmap.org/. [Accessed 19 January 2020].

[21] Rapid7, “Metasploit Penetration Testing Software,” 2020. [Online]. Available: https://www.rapid7.com/products/metasploit/?utm_source=google&utm_medium=cpc&utm_term=&utm_content=410237630346&utm_campaign=brand-golden-keywords&CS=google&gclid=CjwKCAiA1L_xBRA2EiwAgcLKA38yO-h-PRdLAory-zeCpfCj8X0xylmoWWwX2WPaNlr4rtyfHIRZ-hoC-FgQAvD_BwE. [Accessed 19 January 2020].

[22] Fuzz The Planet, “boofuzz 0.0.3,” 2020. [Online]. Available: https://pypi.org/project/boofuzz/0.0.3/. [Accessed 20 January 2020].

[23] “Enable Intel Performance Monitoring Hardware in a Hyper-V virtual machine,” 09 September 2019. [Online]. Available: https://docs.microsoft.com/en-us/windows-server/virtualization/hyper-v/manage/performance-monitoring-hardware. [Accessed 24 January 2020].