ATTACK AND DEFENCE STRATEGIES TO IMPLEMENT END POINT PROTECTION IN CLIENT SYSTEMS

Project Report

Submitted in partial fulfillment of the requirements for the degree of

B.E. (Computer Engineering)
by
Bapu Moraye (Seat No: 0253)
Macbeth Fernandes (Seat No: 0263)
Rahul Hulli (Seat No: 0280)
Sohamkumar Chauhan (Seat No: 0258)

Guide: Prof. Nagaraj Vernekar
Associate Professor

Computer Engineering Department
Goa College of Engineering
(Government of Goa)
Goa University
(2018-2019)

DECLARATION

We, Bapu Moraye, Macbeth Fernandes, Rahul Hulli and Sohamkumar Chauhan declare that the work presented in this project entitled "Attack And Defence Strategies to Implement End Point Protection in Client Systems" submitted to the department of Computer Engineering, Goa College of Engineering, Farmagudi, for the award of the Bachelor's degree in Computer Engineering, is an original work done by us under the guidance of Prof. Nagaraj Vernekar, Associate Professor, Goa College of Engineering. We have neither plagiarized nor submitted the same work for the award of any other degree.

Bapu Moraye
(Roll No: 151105003)
(Seat No: 0253)

Macbeth Fernandes
(Roll No: 151105013)
(Seat No: 0263)

Rahul Hulli
(Roll No: 151105023)
(Seat No: 0280)

Sohamkumar Chauhan
(Roll No: 151105008)
(Seat No: 0258)

DATE:

PLACE: Farmagudi

APPROVAL SHEET

The project entitled "Attack And Defence Strategies to Implement End Point Protection in Client Systems" by

Bapu Moraye (Seat No: 0253)
Macbeth Fernandes (Seat No: 0263)
Rahul Hulli (Seat No: 0280)
Sohamkumar Chauhan (Seat No: 0258)

completed in the year 2018-19 is approved for a fulfillment of the requirements for the degree of BACHELOR OF ENGINEERING in COMPUTER ENGINEERING, and is a record of bonafide work carried out successfully.

Prof. Nagaraj Vernekar (Project Guide) Computer Engineering Department Goa College Of Engineering	Dr. J. A. Laxminarayana (Head of Department) Computer Engineering Department Goa College Of Engineering

Dr. Krupashankara M.S.
(Principal)
Goa College Of Engineering

(Internal Examiner) Name: Designation:	(External Examiner) Name: Designation:

DATE:

PLACE: Farmagudi

ACKNOWLEDGEMENT

We would like to offer our sincere thanks to all those who helped us complete this final year project.We express a great appreciation to the principal of Goa College Of Engineering, **Dr. M.S. Krupashankara**.

We are particularly grateful for the assistance provided by our project guide **Prof. Nagaraj Vernekar**. His constant encouragement and guidance helped us complete our project.

Furthermore, we would like to express our appreciation to **Dr. J. A. Laxminarayana, Dr. Gajanan Gaude, Prof. Manisha Naik Gaonkar** and **Prof. Janhavi Naik** for providing us with valuable suggestions throughout the course of the project.

We acknowledge our seniors and fellow classmates for providing tips and valuable suggestions which truly helped us to excel. Also, we would also like to thank our parents and family members for always being so supportive and inspiring.

Finally, we thank our team-mates for always being positive, persistent and supportive towards each other throughout the span of this project. Good teamwork, cooperation and understanding was key for the successful outcome of this report. We are grateful for having each other.

ABSTRACT

End-Point Protection has been the need of the hour since a very long time in the Cyber World. Various defensive strategies have been devised to implement security features for the benefit of the client systems. Computer Security systems are of many kinds, like Signature-Based Detection, Reputation systems, etc. and now we have entered an era in the cyber time-line where Machine Learning Concepts are also being used to implement security features on client systems. It is observed that client-side systems, having a lot of sensitive data, now focus on installing a number of different security tools which they may or may not really need. This process is being done at most times blindly at the client-side. A lot of security systems lead to a greater amount of work load on the client systems, and may even lead to the systems being heavily focussed more on the security aspect of a computer. The present study is an attempt to testify basic attack strategies that the client systems face most commonly, measure the degree of compromise that the attack achieves, and apply various known defensive strategies so that we can testify the number of defence strategies which are most effective against a particular attack.

Contents

List of Figures

List of Tables

Abbreviations

API - Application Program Interface
EPP - End Point Protection
EPS - End Point Security
EMET - Enhanced Mitigation Experience Toolkit
IDS - Intrusion Detection System
IPS - Intrusion Prevention System
RPO - Recovery Project Objective
RTO - Recovery Time Objective
RISC - Reduced Instruction Set Computer
SET - Social Engineering Toolkit
SRP - Software Restriction Policies
SPEC - Standard Performance Evaluation Corporation
HBF - Host Based Firewall
CPU - Central Processing Unit
DBMS - Database Management System
NMap - Network Map
VLIW - Very Long Instruction Word
WAF - Web Application Firewall
WBF - Web Based Firewall
MFLOPs -Mega Floating-point Operations per second
MIS - Management

Chapter 1

Introduction

The introduction of this project is divided into 5 aspects:

1) Attack and Defence Concept:
Attack and Defence strategy is a strategy developed to ensure and deliver satisfactory security features for particular websites, applications, softwares and even hardware devices. The main rule behind this strategy is simple, there exists a team of adversaries who would perform an attack on a targeted system, and there would correspondingly be a team of defenders who would attempt to defend a system using some defensive strategies either build by them or among the already existing ones. This strategy is commonly used in various industries to ensure product security. This idea has a massive advantage. Suppose say a company, say company A, develops a product, plans and uses certain defensive measures just upon discussion with a set of expert panel members, without checking out by what extent does that defence mechanism protect the privacy of users using the particular product, it may most likely miss out on some major loopholes that might have existed, or even may have generated during the process, hence resulting in the product being vulnerable. But instead, if company A had to implement the security features, by trying out different attack strategies upon the product, analysing the results so obtained, like the amount of data compromise, system issues which get generated during the process, etc., and then introduce different defence strategies to nullify the attack, then they are most likely to cover the generated and existing vulnerabilities atleast to a minimal extent.

2) Security Through Attack And Defence:
Attackers have ramped up their efforts with a dangerous cocktail of social engineering, Web-based attacks and persistence. How will your organization stay ahead? It takes time and money to adjust IT security measures in response to evolving attack tactics. As defenders gradually update their security measures, attackers respond accordingly. Such armsrace dynamics lead to threats of increasing sophistication and efficiency. Today's cybercriminals often have a long-term interest in their targets and often employ social engineering to get inside a protected environment. Their tactics commonly include malicious payload that attempts to compromise the victim's system and may continue spreading within the organization. They also increasingly focus on weaknesses at the application, rather than system or network levels, to obtain data that provide the most value. Defending IT infrastructure involves understanding attack tactics that are particularly effective today. When the system gets attacked, its only then that the developer correctly determines the loophole. If not, the developer may sit under the impression that the system is very safe. The Attack And Defence strategy aims at this very aspect, to intentionally attack a system and check out the amount of compromise through the loophole, and then try to patch the loophole using some defensive strategies which may exist or may be developed and check if the attack is nullified. This process is recursive until the defenders get a set of defensive strategies to more or less, immunise the product against the attack strategies that have been tried out on that product.

3) End-Point Protection:
The term End-Point refers to the client system. This system can be any device, a laptop, a desktop, an android phone, or even a television. Endpoint security or End-Point protection is an approach to the protection of computer networks that are remotely bridged to client devices. The connection of laptops, tablets, mobile phones and other wireless devices to corporate networks creates attack paths for security threats. Endpoint security attempts to ensure that such devices follow a definite level of compliance to standards. Endpoint security management is a software approach which helps to identify and amange the user's computers access over a corporate network. This allows the network administrator to restrict certain website access to specific users in order to maintain and comply with the organization's policies and standards. Endpoint security is becoming a more common IT security function and concern as more employees bring consumer mobile devices to work and companies allow its mobile workforce to use these devices

on the corporate network.

4) Security Problems With Normal Client PCs:
The users of most endpoint systems tend to overburden the operating system with a number of security tools which may or may not be necessary. Although the tools are patches to the loopholes, they are bulky and can cause the system to lag. Moreover, having security patches indicates to any lurking adversary that the system may have sensitive data, thus making it an attack vector (a system which has higher chances of being attacked). Client systems normally tend to simply install antivirus systems and firewalls in response to any type of attack whatsoever. The operating systems (eg. Windows 10) already have defensive systems like Windows Defender, or EMET, SRP etc, which clients may underestimate. Security systems like signature-based detection, which come available with free antivirus softwares, are no longer safe against recent malwares, as tremendous number of variations of a particular malware get generated everyday. Reputation systems are advisories, they only advise whether the softwares are safe and do not stop you from installation. According to a recent event, Kaspersky Security Analyst Summit, held at Cancun, Mexico on March 2018, a topic regarding AI and ML in cyber security being dangerous, was discussed. Machine Learning algorithms are proven to be dangerous time and again in the recent past, plus the strategy taking up a lot of computational procedures and hence slowing down the systems, is also a matter of concern.

5) The Project Concept:
What if most common attacks were tested and analysed on a client system first and then some major defensive strategies were applied, to come up with an optimal set of defensive strategies that can protect the OS? This project aims at this very aspect. The operating system used will be Windows 10. The attacking system will be a Kali Linux OS. 7 different attack strategies will be tried out on the windows system using the Kali Linux OS, the results will be analysed, and then 9 defensive strategies, which are already being used by clients, will be applied, the results will be analysed and a set of defensive strategies will be retrieved.

Chapter 2

Literature Review

2.1 Concept of Benchmarking

- In computing, a benchmark is the act of running a computer program, a set of programs, or other operations, in order to assess the relative performance of an object, normally by running a number of standard tests and trials against it. The term benchmark is also commonly utilized for the purposes of elaborately designed benchmarking programs themselves[6].
- Benchmarking is usually associated with assessing performance characteristics of computer hardware, for example, the floating point operation performance of a CPU, but there are circumstances when the technique is also applicable to software. Software benchmarks are, for example, run against compilers or database management systems (DBMS)[9].
- Benchmarks provide a method of comparing the performance of various subsystems across different chip/system architectures.
- Test suites are a type of system intended to assess the correctness of software.

2.1.1 Purpose

- As computer architecture advanced, it became more difficult to compare the performance of various computer systems simply by looking at their

specifications. Therefore, tests were developed that allowed comparison of different architectures. For example, Pentium 4 processors generally operated at a higher clock frequency than Athlon XP or PowerPCprocessors, which did not necessarily translate to more computational power; a processor with a slower clock frequency might perform as well as or even better than a processor operating at a higher frequency.

- Benchmarks are designed to mimic a particular type of workload on a component or system. Synthetic benchmarks do this by specially created programs that impose the workload on the component. Application benchmarks run real-world programs on the system. While application benchmarks usually give a much better measure of real-world performance on a given system, synthetic benchmarks are useful for testing individual components, like a hard disk or networking device[10].
- Benchmarks are particularly important in CPU design, giving processor architects the ability to measure and make tradeoffs in micro-architectural decisions. For example, if a benchmark extracts the key algorithms of an application, it will contain the performance-sensitive aspects of that application. Running this much smaller snippet on a cycle-accurate simulator can give clues on how to improve performance.
- Prior to 2000, computer and microprocessor architects used SPEC to do this, although SPEC's Unix-based benchmarks were quite lengthy and thus unwieldy to use intact.
- Computer manufacturers are known to configure their systems to give unrealistically high performance on benchmark tests that are not replicated in real usage. For instance, during the 1980s some compilers could detect a specific mathematical operation used in a well-known floating-point benchmark and replace the operation with a faster mathematically equivalent operation. However, such a transformation was rarely useful outside the benchmark until the mid-1990s, when RISC and VLIW architectures emphasized the importance of compiler technology as it related to performance. Benchmarks are now regularly used by compiler companies to improve not only their own benchmark scores, but real application performance.
- CPUs that have many execution units — such as a superscalar CPU, a VLIW CPU, or a reconfigurable computing CPU — typically have slower clock

rates than a sequential CPU with one or two execution units when built from transistors that are just as fast. Nevertheless, CPUs with many execution units often complete real-world and benchmark tasks in less time than the supposedly faster high-clock-rate CPU.

- Given the large number of benchmarks available, a manufacturer can usually find at least one benchmark that shows its system will outperform another system; the other systems can be shown to excel with a different benchmark.
- Manufacturers commonly report only those benchmarks (or aspects of benchmarks) that show their products in the best light. They also have been known to mis-represent the significance of benchmarks, again to show their products in the best possible light. Taken together, these practices are called bench-marketing.
- Ideally benchmarks should only substitute for real applications if the application is unavailable, or too difficult or costly to port to a specific processor or computer system. If performance is critical, the only benchmark that matters is the target environment's application suite[1].

2.1.2 Types of Benchmarking

- Real Program

  Word processing software

  Tool software of CAD

  User's application software (i.e.: MIS)

- Component Benchmark/Microbenchmark

  Core routine consists of a relatively small and specific piece of code.

  Measure performance of a computer's basic components.

  May be used for automatic detection of computer's hardware parameters like number of registers, cache size, memory latency, etc.

- Kernel

  Contains key codes

Normally abstracted from actual program

Popular kernel: Livermore loop

Linpack benchmark (contains basic linear algebra subroutine written in FORTRAN language)

Results are represented in MFLOPS

- Synthetic Benchmark

  Procedure for programming synthetic benchmark:

  1. Take statistics of all types of operations from many application programs
  2. Get proportion of each operation
  3. Write program based on the proportion above

  Types of Synthetic Benchmark are:

  1. Whetstone
  2. Dhrystone

- These were the first general purpose industry standard computer benchmarks. They do not necessarily obtain high scores on modern pipelined computers.
- I/O benchmarks
- Database Benchmarks

  Measure the throughput and response times of database management systems (DBMS)

- Parallel benchmarks

  Used on machines with multiple scores and/or processors, or systems consisting of multiple machines[20]

2.1.3 RealBench Software

- What is Realbench?

  Realbench is a benchmark that uses open source applications and simple scripting to simulate real-world performance of a PC system. It's designed for to show the difference:

  - Before and after a PC upgrade
  - To gauge the real effect of an overclock
  - To gauge the real effect of BIOS tweaks

- Who can use it?

  Anyone! As long as you're running a 64-bit version of Windows, any PC or laptop can run the benchmark to get a score!

- The Tests

  Realbench features several open source software's with the latest CPU extensions, which each test a different part of the system:

  - GIMP Image Editing

    This focuses on single threaded CPU and memory performance, therefore CPU clock speed and memory efficiency (timings + frequency) are the keys to a good score. It uses up to SSE4.2 CPU extensions[15].

  - Handbrake h.264 video compression

    This focuses on multi-threaded CPU and cache performance, therefore the more CPU threads, cache and clock speed you have the better the score. It uses up to AVX CPU extensions.

  - LuxMark rendering

    This focuses entirely on OpenCL performance. It will check for GPU accelerated OpenCL first, before defaulting to CPU if it isn't present. It is also compatible with AMD's upcoming hUMA between APU and GPU. It scales perfectly across all available resources, so the more OpenCL capable GPUs installed the better the score. OpenCL driver efficiency is also

key to this test, with some components performing better than others. The test runs for a fixed period and is calculated on the sustained KSample/sec the system can generate[17].

- Heavy Multitasking

  Test uses a combination of the above tests to simulate a heavy multitasking scenario that loads the entire system.

- Note: All the tests need to be selected in order to get a final score!

2.1.4 HWMonitor

HWMonitor is a hardware monitoring program that reads PC systems main health sensors : voltages, temperatures, fans speed. The program handles the most common sensor chips, like ITE IT87 series, most Winbond ICs, and others. In addition, it can read modern CPUs on-die core thermal sensors, as well has hard drives temperature via S.M.A.R.T, and video card GPU temperature[8].

Benchmarking Parameters

You can find a number of different benchmarking tools online and not one is better than the other. Yet, benchmarking tests do become more complex, depending on what you wish to test.
General Parameters:

A general benchmark will gauge three simple variables: clock speeds, temperatures, and voltages.

HWMonitor provides this exact information. Although using monitoring software does not officially qualify as benchmarking, HWMonitor will allow users to keep track of different readings throughout the benchmarking process. As your GPU and CPU work harder, your temperature readings will rise.

This simple monitoring method displays two key variables: clock speeds and temperature readings. If your temperature readings are high (80-90 °C) at idle — meaning your PC is not working very hard — you should consider taking measures to cool it down. If your components are cool at idle conditions, but increase dramatically under load, your GPU fan may not be working or may be working improperly[11].

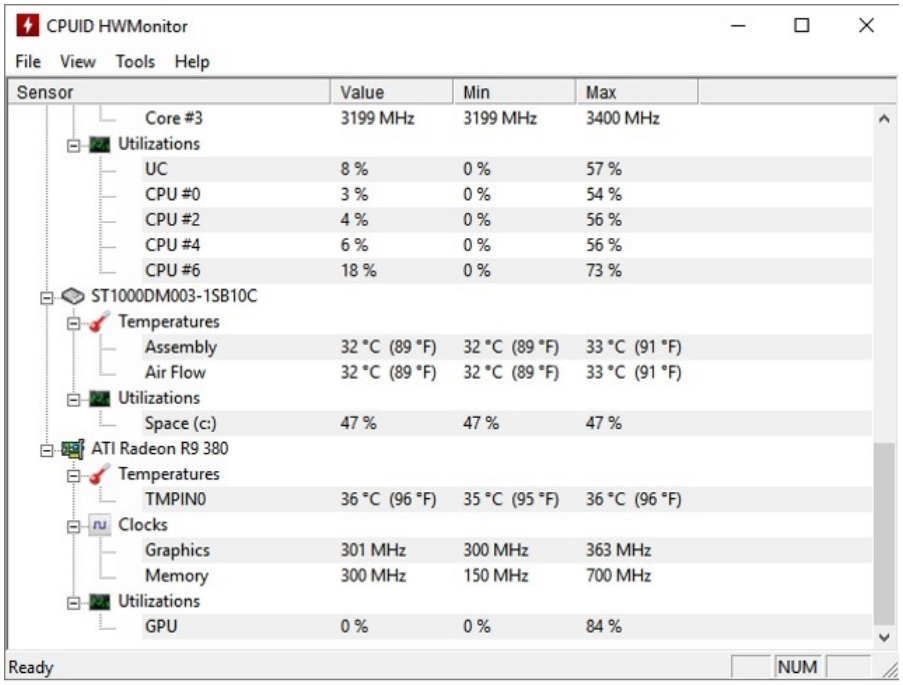

Figure 2.1: Sample Benchmarking Output

2.2 Kali Linux OS

Kali Linux has a dedicated project set aside for compatibility and porting to specific Android devices, called Kali Linux NetHunter.
It is the first Open Source Android penetration testing platform for Nexus devices, created as a joint effort between the Kali community member “BinkyBear” and Offensive Security. It supports Wireless 802.11 frame injection, one-click MANA Evil Access Point setups, HID keyboard (Teensy like attacks), as well as Bad USB MITM attacks.
BackTrack (Kali’s predecessor) contained a mode known as forensic mode, which was carried over to Kali via live boot. This mode is very popular for many reasons, partly because many Kali users already have a bootable Kali USB drive or CD, and this option makes it easy to apply Kali to a forensic job. When booted in forensic mode, the system doesn’t touch the internal hard drive or swap space and auto mounting is disabled. However, the developers recommend that users test these features extensively before using Kali for real world forensics.

2.2.1 Tools

Kali Linux includes security tools, such as:

- Aircrack-ng
- Burp suite
- Cisco Global Exploiter, a hacking tool used to find and exploit vulnerabilities in Cisco
- Network systems
- Ettercap
- John the Ripper
- Kismet
- Maltego
- Metasploit framework
  - Nmap
  - OWASP ZAP
- Social engineering tools.
  - Wireshark
  - Hydra
- Reverse Engineering tools
  - Binwalk
  - Foremost
  - Volatility

These tools can be used for a number of purposes, most of which involve exploiting a victim network or application, performing network discovery, or scanning a target IP address. Many tools from the previous version (Backtrack) were eliminated to focus on the most popular penetration testing applications.

2.3 Footprinting

Footprinting (also known as reconnaissance) is the technique used for gathering information about computer systems and the entities they belong to. To get this information, a hacker might use various tools and technologies. This information is very useful to a hacker who is trying to crack a whole system.

When used in the computer security lexicon, ”Footprinting” generally refers to one of the pre-attack phases; tasks performed prior to doing the actual attack. Some of the tools used for Footprinting are Sam Spade, nslookup, traceroute, Nmap and neotrace.

Techniques used for Footprinting:

- DNS queries
- Network enumeration
- Network queries
- Operating system identification
- Organizational queries
- Ping sweeps
- Point of contact queries
- Port Scanning
- Registrar queries (WHOIS queries)
- SNMP queries
- World Wide Web spidering

2.3.1 Uses of Footprinting

It allows a hacker to gain information about the target system. This information can be used to carry out further attacks on the system. That is the reason by which it may be named a Pre-Attack, since all the information is reviewed in order to get a complete and successful resolution of the attack.

2.3.2 Whois

WHOIS is a web application used to get information about the target website, such as the administrator's e-mail address and details about the registration. WHOIS is a very large database and contains information of approximately all the websites. It can be searched by domain name.

2.3.3 Traceroutes

Information can also be gathered using the command Tracert (”traceroute”), which is used to trace a path between a user and the target system on the networks. That way it becomes clear where a request is being forwarded and through which devices. In Linux systems, the tracepath and traceroute commands are also available for doing traceroute operations.

2.3.4 NMap

Nmap (Network Mapper) is a free and open-source security scanner, originally written by Gordon Lyon (also known by his pseudonym Fyodor Vaskovich), used to discover hosts and services on a computer network, thus building a ”map” of the network. To accomplish its goal, Nmap sends specially crafted packets to the target host(s) and then analyzes the responses.
The software provides a number of features for probing computer networks, including host discovery and service and operating-system detection. These features are extensible by scripts that provide more advanced service detection, vulnerability detection, and other features. Nmap can adapt to network conditions including latency and congestion during a scan. The Nmap user community continues to develop and refine the tool.

Features

Nmap features include:

- Host discovery – Identifying hosts on a network. For example, listing the hosts that respond to TCP and/or ICMP requests or have a particular port open.
- Port scanning – Enumerating the open ports on target hosts.

- Version detection – Interrogating network services on remote devices to determine application name and version number.
- OS detection – Determining the operating system and hardware characteristics of network devices.
- Scriptable interaction with the target – using Nmap Scripting Engine (NSE) and Lua programming language.

Nmap can provide further information on targets, including reverse DNS names, device types, and MAC addresses.

Typical Uses of NMap

- Auditing the security of a device or firewall by identifying the network connections which can be made to, or through it.
- Identifying open ports on a target host in preparation for auditing.
- Network inventory, network mapping, maintenance and asset management.
- Auditing the security of a network by identifying new servers.
- Generating traffic to hosts on a network, response analysis and response time measurement.
- Finding and exploiting vulnerabilities in a network.
- DNS queries and sub-domain search

2.3.5 Zenmap

Zenmap is the official Nmap Security Scanner GUI. It is a multi-platform (Linux, Windows, Mac OS X, BSD, etc.) free and open source application which aims to make Nmap easy for beginners to use while providing advanced features for experienced Nmap users.

Frequently used scans can be saved as profiles to make them easy to run repeatedly. A command creator allows interactive creation of Nmap command lines.

Scan results can be saved and viewed later. Saved scan results can be compared with one another to see how they differ. The results of recent scans are

stored in a searchable database.

Interactive and graphical results viewing – Zenmap can display Nmap's normal output, but you can also arrange its display to show all ports on a host or all hosts running a particular service. It summarizes details about a single host or a com scan in a convenient display. You can even use Zenmap to draw a topology map of discovered networks.

Comparison – you can use Zenmap to graphically show the differences between two scans. This can help you to track new hosts or services appearing on their networks, or existing ones going down

2.4 Phishing Attack

Phishing is the fraudulent attempt to obtain sensitive information such as usernames, password and credit card details (and money), often for malicious reasons, by disguising as a trustworthy entity in an electronic communication. The word is a neologism created as a homophone of fishing due to the similarity of using a bait in an attempt to catch a victim. The annual worldwide impact of phishing could be as high as US $5 billion.

Phishing is typically carried out by email spoofing or instant messaging, and it often directs users to enter personal information at a fake website, the look and feel of which are identical to the legitimate site, the only difference being the URL of the website in concern. Communications purporting to be from social web sites, auction sites, banks, online payment processors or IT administrators are often used to lure victims. Phishing emails may contain links to websites that distribute malware.

What really distinguishes phishing is the form the message takes: the attackers masquerade as a trusted entity of some kind, often a real or plausibly real person, or a company the victim might do business with. It's one of the oldest types of cyberattacks, dating back to the 1990s, and it's still one of the most widespread and pernicious, with phishing messages and techniques becoming increasingly sophisticated.

Some phishing scams have succeeded well enough to make waves:

- Perhaps one of the most consequential phishing attacks in history happened in 2016, when hackers managed to get Hillary Clinton campaign

chair John Podesta to offer up his Gmail password. The ”fappening” attack, in which intimate photos of a number of celebrities were made public, was originally thought to be a result of insecurity on Apple’s iCloud servers, but was in fact the product of a number of successful phishing attempts.

- In 2016, employees at the University of Kansas responded to a phishing email and handed over access to their paycheck deposit information, resulting in them losing pay.
- Phishing is an example of social engineering techniques used to deceive users, and exploits weaknesses in current web security. Attempts to deal with the growing number of reported phishing incidents include legislation, user training, public awareness, and technical security measures.
- How is phishing carried out?
  - Advanced-fee scam

    This common email phishing attack is popularized by the “Nigerian prince” email, where an alleged Nigerian prince in a desperate situation offers to give the victim a large sum of money for a small fee upfront. Unsurprisingly, when the fee is paid, no large sum of money ever arrives. The interesting history is that this type of scam has been occurring for over a hundred years in different forms; it was originally known in the late 1800s as the Spanish Prisoner scam, in which a con artist contacted a victim to pray on their greed and sympathy. The con artist is allegedly trying to smuggle out a wealthy Spanish prisoner, who will reward the victim handsomely in exchange for the money to bribe some prison guards.
  - Account deactivation scam

    By playing off the urgency created in a victim who believes an important account is going to be deactivated, attackers are able to trick some people into handing over important information such as login credentials. Here’s an example: the attacker sends an email that appears to come from an important institution like a bank, and they claim the victim’s bank ac-

count will be deactivated if they do not take action quickly. The attacker will then request the login and password to the victim's bank account in order to prevent the deactivation. In a clever version of the attack, once the information is entered, the victim will be directed to the legitimate bank website so that nothing looks out of place.

  - Website forgery scam

    This type of scam is commonly paired with other scams such as the account deactivation scam. In this attack, the attacker creates a website that is virtually identical to the legitimate website of a business the victim uses, such as a bank. When the user visits the page through whatever means, be it an email phishing attempt, a hyperlink inside a forum, or via a search engine, the victim reaches a website which they believe to be the legitimate site instead of a fraudulent copy. All information entered by the victim is collected for sale or other malicious use.

- Spear Phishing

  Spear phishing is an email or electronic communications scam targeted towards a specific individual, organization or business. Although often intended to steal data for malicious purposes, cybercriminals may also intend to install malware on a targeted user's computer.

  This is how it works: An email arrives, apparently from a trustworthy source, but instead it leads the unknowing recipient to a bogus website full of malware. These emails often use clever tactics to get victims' attention. For example, the FBI has warned of spear phishing scams where the emails appeared to be from the National Center for Missing and Exploited Children.

  Many times, government-sponsored hackers and hacktivists are behind these attacks. Cybercriminals do the same with the intention to resell confidential data to governments and private companies. These cybercriminals employ individually designed approaches and social engineering techniques to effectively personalize messages and websites. As a result, even high-ranking targets within

organizations, like top executives, can find themselves opening emails they thought were safe. That slip-up enables cybercriminals to steal the data they need in order to attack their networks.

The act of spear-phishing may sound simple, but spear-phishing emails have improved within the past few years and are now extremely difficult to detect without prior knowledge on spear-phishing protection. Spear-phishing attackers target victims who put personal information on the internet. They might view individual profiles while scanning a social networking site. From a profile, they will be able to find a person's email address, friends list, geographic location, and any posts about new gadgets that were recently purchased. With all of this information, the attacker would be able to act as a friend or a familiar entity and send a convincing but fraudulent message to their target.

To increase success rates, these messages often contain urgent explanations on why they need sensitive information. Victims are asked to open a malicious attachment or click on a link that takes them to a spoofed website where they are asked to provide passwords, account numbers, PINs, and access codes. An attacker posing as a friend might ask for usernames and passwords for various websites, such as Facebook, so that they would be able to access posted photos. In reality, the attackers will use that password, or variations of it, to access different websites that have confidential information such as credit card details or Social Security Numbers. Once criminals have gathered enough sensitive information, they can access bank accounts or even create a new identity using their victim's information. Spear-phishing can also trick people into downloading malware or malicious codes after people click on links or open attachments provided in messages.

2.5 What is a Payload?

A payload refers to the component of a computer virus that executes a malicious activity. Apart from the speed in which a virus spreads, the threat level of a virus is calculated by the damages it causes. Viruses with more powerful payloads tend to be more harmful.

Although not all viruses carry a payload, a few payloads are considered extremely dangerous. Some of the examples of payloads are data destruction, offensive messages and the delivery of spam emails through the infected user's account.

A payload is also known as a destructive payload.

Some viruses just copy themselves from one computer to other. Other viruses may steal data or files, permit eavesdropping or unauthorized access, destroy data and cause other consequences. It is also possible for a virus to carry multiple payloads.

Present-day malware is less likely to incorporate a payload that causes damage to system files; instead, they enable backdoor access to a user's computer and the theft of sensitive information.

Some of the ways to execute a payload include:

- By using an unprotected computer (computer without an anti-virus installed) connected to a network
- By booting the computer using an infected removable medium
- By opening an infected file
- By executing an infected program
- By activating a logic bomb

2.6 SE Toolkit for Kali Linux

The Social-Engineer Toolkit (SET) is an open-source penetration testing framework designed for social engineering. SET has a number of custom attack vectors that allow you to make a believable attack in a fraction of time. These kind of tools use human behaviours to trick them to the attack vectors. SET is a product of TrustedSec, LLC – an information security consulting firm located in Cleveland, Ohio.

2.7 Backdoor

Deploying a backdoor on a computer of an employee of an organization, or victim to gain unauthorized access to the private network.
It is not all about creating a backdoor on IoT devices. Some other types of IoT attacks include:

- Eavesdropping
- Sybil Attack
- Exploit Kits
- Man-in-the-Middle Attack
- Replay Attack
- Forged Malicious Devices
- Side Channel Attack
- Ransomware Attack

2.8 Trojan Concept

Trojan Horse and Trojan are the malicious programs which mislead from its actual intentions. This term is actually derived from a Greek story of a great Wooden horse. This horse had soldiers hiding inside waiting to enter into the city. As this wooden horse reached in the city, soldiers came out and attacked the city.

With this philosophy, Trojan software misleads from its true intentions and wait for best time to attack. These Trojan may provide access to personal information, as well as unauthorized access to the attacker.

The trojan can also lead to infection of other connected devices across a network. Trojan A Malicious Program misleading the user about its actual intention is classified as Trojan.

Trojans are typically spread by Social Engineering. The purpose or most common use of Trojan programs are:

- Creating back door

- Gaining Unauthorized Access
- Steal Information
- Infect Connected Devices
- Ransomware Attacks
- Using Victim for Spamming
- Using Victim as Botnet
- Downloading other malicious software
- Disabling Firewalls

2.9 End Point Security

The term End-Point refers to the client system. This system can be any device, a laptop, a desktop, an android phone, or even a television. Endpoint security or End-Point protection is an approach to the protection of computer networks that are remotely bridged to client devices. The connection of laptops, tablets, mobile phones and other wireless devices to corporate networks creates attack paths for security threats. Endpoint security attempts to ensure that such devices follow a definite level of compliance to standards. Endpoint security management is a software approach which helps to identify and amange the user's computers access over a corporate network. This allows the network administrator to restrict certain website access to specific users in order to maintain and comply with the organization's policies and standards. Endpoint security is becoming a more common IT security function and concern as more employees bring consumer mobile devices to work and companies allow its mobile workforce to use these devices on the corporate network.

Based on the paper presented by Simon Hansman, Ray Hunt [ref], computer and network attacks was needed to be categorized so as to improve computer security and an easy descriptions of attacks. They reasoned that when attacks would be categorized, it would be easier to properly distinguish and highlight vulnerable points or loopholes and accordingly patch them. They proposed a taxonomy which consisted of four dimensions which would provide

a holistic taxonomy in order to deal with inherent problems of the computer and network attacks.

Paper by Xinyue Yang [ref] pointed out some major attack tools used by attackers, namely D.O.S. attacks, BackOrfice, Glacier, NetSpy, KeyboardGhost, ExeBind, etc. The paper also mentioned various types of attacks and defence practices introduced.

Paper by Mark C. Dacier [ref] defined security challenges faced in Software-Defined Networking (SDN)

Paper by Yang Sun [ref] used the concept of Parento Optimization for the Analysis of Network And Defence Strategies.

2.10 Detect, Prevent, Protect

A bank would never leave its assets inside an unguarded safe alone. Typically, access to the safe requires passing through layers of protection that might include human guards and locked doors with special access controls. Furthermore, the room where the safe resides could be monitored by closed-circuit television, motion sensors, and alarm systems that can quickly detect unusual activity. The sound of an alarm might trigger the doors to automatically lock, the police to be notified, or the room to fill with tear gas. Layered security, as in the previous example, is known as defence in depth. This security is implemented in overlapping layers that provide the three elements needed to secure assets: prevention, detection, and response. defence in depth also seeks to offset the weaknesses of one security layer by the strengths of two or more layers. In the information security world, defence in depth requires layering security devices in a series that protects, detects, and responds to attacks on systems. For example, a typical Internet-attached network designed with security in mind includes routers, firewalls, and intrusion detection systems (IDS) to protect the network from would-be intruders; employs traffic analyzers and real-time human monitors who watch for anomalies as the network is being used to detect any breach in the layers of protection; and relies on automated mechanisms to turn off access or remove the system from the network in response to the detection of an intruder. Finally, the security of each of these mechanisms must be thoroughly tested before deployment to ensure that the integrated system is suitable for normal operations. After all, a chain is only as good as its weakest link.

2.11 Reputation Systems

Reputation systems are programs that allow users to rate each other in online communities in order to build trust through reputation. Some common uses of these systems can be found on Ecommerce websites such as eBay, Amazon.com, and Etsy as well as online advice communities such as Stack Exchange. These reputation systems represent a significant trend in ”decision support for Internet mediated service provisions”. With the popularity of online communities for shopping, advice, and exchange of other important information, reputation systems are becoming vitally important to the online experience. The idea of reputations systems is that even if the consumer can’t physically try a product or service, or see the person providing information, that they can be confident in the outcome of the exchange through trust built by recommender systems. Collaborative filtering, used most commonly in recommender systems, are related to reputation systems in that they both collect ratings from members of a community. The core difference between reputation systems and collaborative filtering is the ways in which they use user feedback. In collaborative filtering, the goal is to find similarities between users in order to recommend products to customers. The role of reputation systems, in contrast, is to gather a collective opinion in order to build trust between users of an online community.

2.11.1 Types

- Online
  Howard Rheingold states that online reputation systems are ”computer-based technologies that make it possible to manipulate in new and powerful ways an old and essential human trait”. Rheingold says that these systems arose as a result of the need for Internet users to gain trust in the individuals they transact with online. The trait he notes in human groups is that social functions such as gossip ”keeps us up to date on who to trust, who other people trust, who is important, and who decides who is important”. Internet sites such as eBayand Amazon, he argues, seek to make use of this social trait and are ”built around the contributions of millions of customers, enhanced by reputation systems that police the quality of the content and transactions exchanged through the site”.
- Reputation Banks
  The emerging sharing economy increases the importance of trust in peer-to-

peer marketplaces and services. Users can build up reputation and trust in individual systems but usually don't have the ability to carry those reputations to other systems. Rachel Botsman and Roo Rogers argue in their book What's Mine is Yours (2010), that "it is only a matter of time before there is some form of network that aggregates reputation capital across multiple forms of Collaborative Consumption". These systems, often referred to as reputation banks, try to give users a platform to manage their reputation capital across multiple systems.

2.11.2 Attack and Defence

Reputation systems are in general vulnerable to attacks, and many types of attacks are possible. As the reputation system tries to generate an accurate assessment based on various factors including but not limited to unpredictable user size and potential adversarial environments, the attacks and defence mechanisms play an important role in the reputation systems. Attack classification of reputation system is based on identifying which system components and design choices are the targets of attacks. While the defence mechanisms are concluded based on existing reputation systems.

2.11.3 Defence Strategies

Here are some strategies to prevent the above attacks:

- Preventing Multiple Identities
- Mitigating Generation of False Rumours
- Mitigating Spreading of False Rumours
- Preventing Short-Term Abuse of the System
- Mitigating Denial of Service Attacks

2.12 Blacklists

In computing, a blacklist or block list is a basic access control mechanism that allows through all elements (email addresses, users, passwords, URLs, IP addresses, domain names, file hashes, etc.), except those explicitly mentioned. Those items

on the list are denied access. The opposite is a whitelist, which means only items on the list are let through whatever gate is being used. A greylist contains items that are temporarily blocked (or temporarily allowed) until an additional step is performed. Blacklists can be applied at various points in a security architecture, such as a host, web proxy, DNS servers, email server, firewall, directory servers or application authentication gateways. The type of element blocked is influenced by the access control location. DNS servers may be well-suited to block domain names, for example, but not URLs. A firewall is well-suited for blocking IP addresses, but less so for blocking malicious files or passwords. Example uses include a company that might prevent a list of software from running on its network, a school that might prevent access to a list of web sites from its computers, or a business that wants to ensure their computer users are not choosing easily guessed, poor passwords.

2.12.1 Example Systems to Protect

Blacklists are used to protect a variety of systems in computing. The content of the blacklist is likely needing to be targeted to the type of system defended.

- Information Systems
  An information system includes end-point hosts like user machines and servers. A blacklist in this location may include certain types of software that are not allowed to run in the company environment. For example, a company might blacklist peer to peer file sharing on its systems. In addition to software, people, devices and Web sites can also be blacklisted.
- Email
  Most email providers have a anti-spam feature that essentially blacklists certain email addresses if they are deemed unwanted. How this happens is when a successful phishing attack (from an address that is forged from reliable accounts to try to recover personal information) is executed, then the email device deems the address to be spam, and proceeds to blacklist the address. An e-mail spam filter may keep a blacklist of email addresses, any mail from which would be prevented from reaching its intended destination. It may also use sending domain names or sending IP addresses to implement a more general block. In addition to private email blacklists, there are lists that are kept for public use, examples are:
  - China Anti-Spam Alliance

  - Fabel Spamsources
  - Spam and Open Relay Blocking System
  - The DrMX Project

- Web browsing
  The goal of a blacklist in a web browser is to prevent the user from visiting a malicious or deceitful web page via filtering locally. A common web browsing blacklist is Google's Safe Browsing, which is installed by default in Firefox, Safari, and Chrome.

- Usernames and Passwords
  Blacklisting can also apply to user credentials. It is common for systems or websites to blacklist certain reserved usernames that are not allowed to be chosen by the system or website's user populations. These reserved usernames are commonly associated with built-in system administration functions. Password blacklists are very similar to username blacklists but typically contain significantly more entries than username blacklists. Password blacklists are applied to prevent users from choosing passwords that are easily guessed or are well known and could lead to unauthorized access by malicious parties. Password blacklists are deployed as an additional layer of security, usually in addition to a password policy, which sets the requirements of the password length and/or character complexity. This is because there are a significant number of password combinations that fulfil many password policies but are still easily guessed (i.e., Password123, Qwerty123).

- Considerations of Usage
  As expressed in a recent conference paper focusing on blacklists of domain names and IP addresses used for Internet security, "these lists generally do not intersect. Therefore, it appears that these lists do not converge on one set of malicious indicators."This concern combined with an economic model means that, while blacklists are an essential part of network defence, they need to be used in concert with whitelists and greylists. An example would be the Adblock Plus blocklist that includes a number of features including whitelists within the blacklist by adding a prefix of two at symbols and two pipe symbols e.g. "@@|| www.blocksite.com".

- Intrusion Prevention Systems
  An Intrusion Prevention System (IPS) is a network security/threat preven-

tion technology that examines network traffic flows to detect and prevent vulnerability exploits. Vulnerability exploits usually come in the form of malicious inputs to a target application or service that attackers use to interrupt and gain control of an application or machine. Following a successful exploit, the attacker can disable the target application (resulting in a denial-of-service state), or can potentially access to all the rights and permissions available to the compromised application.

- Prevention
  The IPS often sits directly behind the firewall and provides a complementary layer of analysis that negatively selects for dangerous content. Unlike its predecessor the Intrusion Detection System (IDS)—which is a passive system that scans traffic and reports back on threats—the IPS is placed inline (in the direct communication path between source and destination), actively analyzing and taking automated actions on all traffic flows that enter the network. Specifically, these actions include sending an alarm to the administrator (as would be seen in an IDS), dropping the malicious packets and blocking traffic from the source address and resetting the connection. As an inline security component, the IPS must work efficiently to avoid degrading network performance. It must also work fast because exploits can happen in near real-time. The IPS must also detect and respond accurately, so as to eliminate threats and false positives (legitimate packets misread as threats).

- Detection
  The IPS has a number of detection methods for finding exploits, but signature-based detection and statistical anomaly-based detection are the two dominant mechanisms.Signature-based detection is based on a dictionary of uniquely identifiable patterns (or signatures) in the code of each exploit. As an exploit is discovered, its signature is recorded and stored in a continuously growing dictionary of signatures. Signature detection for IPS breaks down into two types:

  * Exploit-facing signatures identify individual exploits by triggering on the unique patterns of a particular exploit attempt. The IPS can identify specific exploits by finding a match with an exploit-facing signature in the traffic stream.

  * Vulnerability-facing signatures are broader signatures that target

the underlying vulnerability in the system that is being targeted. These signatures allow networks to be protected from variants of an exploit that may not have been directly observed in the wild, but also raise the risk of false positives. Statistical anomaly detection takes samples of network traffic at random and compares them to a precalculated baseline performance level. When the sample of network traffic activity is outside the parameters of baseline performance, the IPS takes action to handle the situation.IPS was originally built and released as a standalone device in the mid-2000s. This however, was in the advent of today's implementations, which are now commonly integrated into Unified Threat Management (UTM) solutions (for small and medium size companies) and next-generation firewalls (at the enterprise level).

2.13 Disk Encryption

Disk encryption is a technology which protects information by converting it into unreadable code that cannot be deciphered easily by unauthorized people. Disk encryption uses disk encryption software or hardware to encrypt every bit of data that goes on a disk or disk volume. It is used to prevent unauthorized access to data storage. Expressions full disk encryption (FDE) or whole disk encryption signify that everything on disk is encrypted, but the master boot record (MBR), or similar area of a bootable disk, with code that starts the operating system loading sequence, is not encrypted. Some hardware-based full disk encryption systems can truly encrypt an entire boot disk, including the MBR.

2.13.1 Security Concerns

Most full disk encryption schemes are vulnerable to a cold boot attack, whereby encryption keys can be stolen by coldbooting a machine already running an operating system, then dumping the contents of memory before the data disappears. The attack relies on the data remanence property of computer memory, whereby data bits can take up to several minutes to degrade after power has been removed. Even a Trusted Platform Module (TPM) is not effective against the attack, as the operating system needs to hold the decryption keys in memory in order to access the disk. Full disk encryption is also vulnerable when a computer is stolen when suspended. As wake-up does not involve a BIOS boot sequence, it typically does

not ask for the FDE password. Hibernation, in contrast goes via a BIOS boot sequence, and is safe. All software-based encryption systems are vulnerable to various side channel attacks such as acoustic cryptanalysis and hardware keyloggers. In contrast, self-encrypting drives are not vulnerable to these attacks since the hardware encryption key never leaves the disk controller. Also, most of full disk encryption schemes don't protect from data tampering (or silent data corruption, i.e. bitrot). That means they only provide privacy, but not integrity. Block cipherbased encryption modes used for full disk encryption are not authenticated encryption themselves because of concerns of the storage overhead needed for authentication tags. Thus, if tampering would be done to data on the disk, the data would be decrypted to garbled random data when read and hopefully errors may be indicated depending on which data is tampered with (for the case of OS metadata – by the file system; and for the case of file data – by the corresponding program that would process the file). One of the ways to mitigate these concerns, is to use file systems with full data integrity checks via checksums (like Btrfs or ZFS) on top of full disk encryption. However, cryptsetup started experimentally to support authenticated encryption.

2.13.2 Full Disk Encryption

Benefits

Full disk encryption has several benefits compared to regular file or folder encryption, or encrypted vaults. The following are some benefits of disk encryption:

- Nearly everything including the swap space and the temporary files is encrypted. Encrypting these files is important, as they can reveal important confidential data. With a software implementation, the bootstrapping code cannot be encrypted however. For example, BitLocker Drive Encryption leaves an unencrypted volume to boot from, while the volume containing the operating system is fully encrypted.
- With full disk encryption, the decision of which individual files to encrypt is not left up to users' discretion. This is important for situations in which users might not want or might forget to encrypt sensitive files.
- Immediate data destruction, such as simply destroying the cryptographic keys (crypto-shredding), renders the contained data useless. However, if security towards future attacks is a concern, purging or physical destruction is advised.

The Boot-Key Problem

One issue to address in full disk encryption is that the blocks where the operating system is stored must be decrypted before the OS can boot, meaning that the key has to be available before there is a user interface to ask for a password. Most Full Disk Encryption solutions utilize Pre-Boot Authentication by loading a small, highly secure operating system which is strictly locked down and hashed versus system variables to check for the integrity of the Pre-Boot kernel. Some implementations such as BitLocker Drive Encryption can make use of hardware such as a Trusted Platform Module to ensure the integrity of the boot environment, and thereby frustrate attacks that target the boot loader by replacing it with a modified version. This ensures that authentication can take place in a controlled environment without the possibility of a bootkit being used to subvert the pre-boot decryption. With a pre-boot authentication environment, the key used to encrypt the data is not decrypted until an external key is input into the system.
Solutions for storing the external key include:

- Username / password
- Using a smartcard in combination with a PIN
- Using a biometric authentication method such as a fingerprint
- Using a dongle to store the key, assuming that the user will not allow the dongle to be stolen with the laptop or that the dongle is encrypted as well
- Using a boot-time driver that can ask for a password from the user
- Using a network interchange to recover the key, for instance as part of a PXE boot
- Using a TPM to store the decryption key, preventing unauthorized access of the decryption key or subversion of the boot loader
- Using a combination of the above All these possibilities have varying degrees of security; however, most are better than an unencrypted disk.

2.14 Host Based Firewalls

A host-based firewall is basically firewall software running on a PC or file server. When running on a personal PC, this commonly is called a personal firewall. This

typically is used to enhance your security solution or to provide additional protection to your desktop. You can use literally dozens of free, shareware, and premium software-based firewall products to protect Microsoft Wndows, Apple Macintosh, and the many flavors of UNIX that you might have running as your operating system. Host-based firewalls usually do not have the same capabilities of the other firewall categories that I already discussed. They are typically a simplified packetfiltering firewall that filter on the IP protocol, such as TCP or UDP, source and destination IP addresses, and protocol information such as TCP or UDP port numbers. Hostbased firewalls were built to provide a low-cost alternative to other firewall categories yet still provide a decent level of protection: They typically have fewer filtering and logging capabilities.

2.15 Intrusion Detection Systems

An intrusion detection system (IDS) is a device or software application that monitors a network or systems for malicious activity or policy violations. Any malicious activity or violation is typically reported either to an administrator or collected centrally using a security information and event management (SIEM) system. A SIEM system combines outputs from multiple sources, and uses alarm filtering techniques to distinguish malicious activity from false alarms. [citation needed] IDS types range in scope from single computers to large networks.[1] The most common classifications are network intrusion detection systems (NIDS) and host-based intrusion detection systems (HIDS). A system that monitors important operating system files is an example of an HIDS, while a system that analyses incoming network traffic is an example of an NIDS. It is also possible to classify IDS by detection approach: the most well-known variants are signature-based detection (recognizing bad patterns, such as malware); and anomaly-based detection (detecting deviations from a model of ”good” traffic, which often relies on machine learning). Some IDS products have the ability to respond to detected intrusions. Systems with response capabilities are typically referred to as an intrusion prevention system.

2.15.1 Comparison with Firewall

Although they both relate to network security, an IDS differs from a firewall in that a firewall looks outwardly for intrusions in order to stop them from happening. Firewalls limit access between networks to prevent intrusion and do not signal

an attack from inside the network. An IDS describes a suspected intrusion once it has taken place and signals an alarm. An IDS also watches for attacks that originate from within a system. This is traditionally achieved by examining network communications, identifying heuristics and patterns (often known as signatures) of common computer attacks, and taking action to alert operators. A system that terminates connections is called an intrusion prevention system, and is another form of an application layer firewall.

2.15.2 Limitations

- Noise can severely limit an intrusion detection system's effectiveness. Bad packets generated from software bugs, corrupt DNS data, and local packets that escaped can create a significantly high false-alarm rate.
- It is not uncommon for the number of real attacks to be far below the number of false-alarms. Number of real attacks is often so far below the number of false-alarms that the real attacks are often missed and ignored.
- Many attacks are geared for specific versions of software that are usually outdated. A constantly changing library of signatures is needed to mitigate threats. Outdated signature databases can leave the IDS vulnerable to newer strategies.
- For signature-based IDS, there will be lag between a new threat discovery and its signature being applied to the IDS. During this lag time, the IDS will be unable to identify the threat.
- It cannot compensate for weak identification and authentication mechanisms or for weaknesses in network protocols. When an attacker gains access due to weak authentication mechanisms then IDS cannot prevent the adversary from any malpractice.
- Encrypted packets are not processed by most intrusion detection devices. Therefore, the encrypted packet can allow an intrusion to the network that is undiscovered until more significant network intrusions have occurred.
- Intrusion detection software provides information based on the network address that is associated with the IP packet that is sent into the network. This is beneficial if the network address contained in the IP packet is accurate. However, the address that is contained in the IP packet could be faked or scrambled.

- Due to the nature of NIDS systems, and the need for them to analyse protocols as they are captured, NIDS systems can be susceptible to the same protocol-based attacks to which network hosts may be vulnerable. Invalid data and TCP/IP stack attacks may cause a NIDS to crash.

2.16 Web Application Firewalls

A Web application firewall (WAF) is a firewall that monitors, filters or blocks data packets as they travel to and from a Web application. A WAF can be either network-based, host-based or cloud-based and is often deployed through a proxy and placed in front of one or more Web applications. Running as a network appliance, server plug-in or cloud service, the WAF inspects each packet and uses a rule base to analyze Layer 7 web application logic and filter out potentially harmful traffic.
Web application firewalls are a common security control used by enterprises to protect Web applications against zero-day exploits, impersonation and known vulnerabilities and attackers. Through customized inspections, a WAF is also able to prevent cross-site scripting (XSS) attacks, SQL injection attacks, session hijacking and buffer overflows, which traditional network firewalls and other intrusion detection systems may not be capable of doing. WAFs are especially useful to companies that provide products or services over the Internet.

2.17 OSQuery

OSQuery is an operating system instrumentation framework for Windows, OS X (macOS), Linux, and FreeBSD. The tools make low-level operating system analytics and monitoring both performant and intuitive.
OSQuery exposes an operating system as a high-performance relational database. This allows you to write SQL queries to explore operating system data. With osquery, SQL tables represent abstract concepts such as running processes, loaded kernel modules, open network connections, browser plugins, hardware events or file hashes.

2.18 Disaster Recovery

Disaster recovery (DR) is an area of security planning that aims to protect an organization from the effects of significant negative events. DR allows an organization to maintain or quickly resume mission-critical functions following a disaster.
A disaster can be anything that puts an organization's operations at risk, from a cyberattack to equipment failures to natural disasters. The goal with DR is for a business to continue operating as close to normal as possible. The disaster recovery process includes planning and testing, and may involve a separate physical site for restoring operations.

2.18.1 The Importance of Disaster Recovery: RPO and RTO

- As businesses have become more reliant on high availability, the tolerance for downtime has decreased.
- A disaster can have a devastating effect on a business. Studies have shown that many businesses fail after experiencing a significant data loss, but DR can help.
- Recovery point objective (RPO) and recovery time objective (RTO) are two important measurements in disaster recovery and downtime.
- RPO is the maximum age of files that an organization must recover from backup storage for normal operations to resume after a disaster. The recovery point objective determines the minimum frequency of backups. For example, if an organization has an RPO of four hours, the system must back up at least every four hours.
- RTO is the maximum amount of time, following a disaster, for an organization to recover files from backup storage and resume normal operations. In other words, the recovery time objective is the maximum amount of downtime an organization can handle. If an organization has an RTO of two hours, it cannot be down for longer than that.
- The RPO and RTO help administrators choose optimal disaster recovery strategies, technologies and procedures.
- Meeting tighter RTO windows requires positioning secondary data so that it can be accessed faster. Recovery-in-place is one method of restoring data more quickly. This technology moves backup data to a live state on the

backup appliance, eliminating the need to move data across a network. It can protect against storage system and server failure. Before using recovery-in-place, an organization needs to consider the performance of the disk backup appliance, the time needed to move data from a backup state to a live state, and failback. Since recovery-in-place can take up to 15 minutes, an organization may need to perform replication if it wants a quicker recovery time.

- Preparing for a disaster requires a comprehensive approach that encompasses hardware and software, networking equipment, power, connectivity and testing that ensures DR is achievable within RTO and RPO targets. While implementing a thorough DR plan isn't a small task, the potential benefits are significant.

2.19 Remediation

What does remediation mean? If you look up the root word 'remedy,' you'll see it's defined as "a treatment for an injury or disease," or "a means of eliminating or counteracting something that's undesirable." In terms of cybersecurity incident response, remediation means addressing a breach in the most effective way possible to limit the amount of damage that can potentially be done to the organization being targeted. In reality, cybersecurity involves so much more.
Applying this to cybersecurity incident response, the best approach should dig deeper to find and eradicate the actual cause of the underlying threat, such as locating the malware and other malicious files that caused the breach. Without this extra step, your organization is left vulnerable to the virtually immeasurable damages that can be caused if the true issue isn't taken care of properly.
To truly remediate a cybersecurity incident, you must first identify it and gather as much relevant information about it as possible. That information must then be adequately analysed to determine what type of threat you're dealing with and its potential impact. To give you an idea of what type of 'relevant' information we're talking about, start with the following:

- What systems have been affected?
- Which process is allowing the issue to continue?
- What are the characteristics of the incident?

Only when you have a clear and accurate understanding of what you're up against can you properly address and remediate it. It can be helpful to think of cybersecurity incident response as a process rather than a specific solution. The fact is, today's cyber threats are evolving and becoming more dynamic and complex by the day. Simply preparing in advance for possible scenarios isn't enough anymore. Current day cyber-attacks require immediate response.
Effective cybersecurity incident response cannot be static. It must adapt alongside the changing threat landscape. It requires deep research and data analysis in every step of the process. In other words, it requires a certain degree of intelligence. That's where automation comes into play. The right automated cybersecurity incident response plan should leverage advanced technology, such as machine learning, that will both address the need for round-the-clock monitoring and response as well as adapt intelligently over time.

2.20 Roll-Back

This a feature which when the system is compromised by attacks then the process of reverting the system back to the initial state before it was attacked is called Roll back action.The difference with respect to backup is that in back up we directly start the system recovery from the backup itself but in case of roll back action you are reverting from the compromised state of the system to the stable state.
A strategic disaster recovery plan is based on your organization's data recovery needs. Administrators must decide what is most important: the ability to reproduce the original data, or to reproduce the data in its state just prior to the corruption, or both. The answer will depend on the type of data your organization collects and the impact of its loss.

Chapter 3

Project Objectives

3.1 Project Objectives

- For this project, we will be using two systems, i.e an attacking system and a defensive system.
- Attacking System: Kali Linux System.
- Defensive System: Microsoft Windows 10 Pro.
- This project is divided into 4 phases/milestones.
    1. Normal Benchmarking and Analysis
    2. Attack Phase
    3. Defence Phase
    4. Implementation of API

3.2 Project Methodology

The approach used is called Attack and Defense.

In this project, the process will be divided into 3 steps:

1. The Attack Procedure : Various attacking strategies will be used to attack a given client system. About 5-7 different computer attacks will be done, and

atleast 5-7 different types of viruses will be injected onto the client system.

2. The Defense Procedure : Based on each attack, different defensive strategies will be used, 7-10 in total.
3. The Report : The analysis of every defensive strategy used will be done.

For the success of the project, three domains will be throughly addressed:

- Computer Forensics - To understand different changes that can possibly happen once a system is affected by malware.
- End Point Security - The study of deploying/developing/installing security features onto a client system.
- Computer Hacking - The science of accessing any computer(s) as an adversary without the client's wish.

Chapter 4

Design

4.1 Conceptual Design

The approach used is called Attack and Defence.
In this project, the process will be divided into 3 steps:

1. The Attack Procedure :
   Various attacking strategies will be used to attack a given client system. About 5-7 different computer attacks will be done, and atleast 5-7 different types of viruses will be injected onto the client system.
2. The Defence Procedure :
   Based on each attack, different defensive strategies will be used, 7-10 in total.
3. The Report :
   The analysis of every defensive strategy used will be done.

For the success of the project, three domains will be thoroughly addressed:

- Computer Forensics - To understand different changes that can possibly happen once a system is affected by malware.
- End Point Security - The study of deploying/developing/installing security features onto a client system.

- Computer Hacking - The science of accessing any computer(s) as an adversary without the client's wish.

4.2 Detailed Design

4.2.1 Normal Benchmarking and Analysis

- This is done on the windows system (defensive system).
- End Point Security has become one of the most important sub domains in the computer security domain. This is so, primarily because many users believe that the solution to fighting computer virus is simply installing an antivirus system. The project intends to deliver a set of defensive startegies that would be useful of some popular attacks. The approach would have been to quantify every attack and defence strategy, but this has its own shortcomings:
  - not all problems can be properly quantified, i.e., in some cases, cost classification is possible, but not quantification.
  - quantitative cost estimation is well suited for intrusion detection systems (IDS), but underperforms when choosing a defence strategy.
- Hence, the team decided to quantify basic parameters using a normal benchmarking analysis. The parameters which we would be interested in to get through the project are as follows:
  - Average Temperature for each core.
  - Average CPU Utilization for each core.
  - Average Graphics Utilization.
  - Average Clock Speed for each core.
  - Average Clock Speed for Graphics.
  - Average Graphics Memory Utilization.

To do this, we will be using a benchmarking software called HWMonitor, described in the literature survey of this document.
In this phase, all the applications and multimedia softwares running onto the system will be shut, some of such applications which run at the background will also

be shut through the basic task manager installed onto the windows system. Then the software HWMonitor is opened, which gives a list of parameters and their values at that instant. The parameters are temperatures, powers, utilizations (CPU and memory), clocks, HD Graphics clocks and utilizations (CPU and memory), current voltage, capacities, wear and charge levels and bandwidth. To monitor and derive a graph of the parameters, we need to go to the task bar, click on the option Tools $\longrightarrow$ Logs $\longrightarrow$ Start Recording, or simply click F5. Record it for the amount of time you want to record, here we will do it for 10 seconds, and then go to task bar, click on the option Tools $\longrightarrow$ Logs $\longrightarrow$ Stop Recording, or simply click F5. Automatically a folder opens which has the bmp image files which represent the graphs of the parameters you want to track.
We will use these graphs to calculate the above mentioned parameters.
Some additional parameters that we need to calculate (which we will require more in the Attack Phase) are as follows:

- Capture Cost(cc)
  The capture cost is the cost associated with compromising a device, effectively replacing it with a malicious insider from which the attacker can launch a new attack. This is normally not quantifiable, hence we will measure this in terms of time taken to generate and send a payload plus the time within which the system gets compromised/accessible to the user. During the Phase 1, capture cost will be calculated via a ping, issued through the Kali Linux system.

- Available Connections (conn)
  The available connections are the number of connections that the vertex can be accessed by. Here, the number of connections will be assumed directly proportional to the number of ports of the client system which are open.

- Access (access)
  The access attribute defines whether a vertex can be capture by an attacker. This is either true or false. Just to be fair, we will be using another Benchmarking software, called RealBench Bechmarking Tool, by Asus. This software was chosen since it does a rigorous benchmarking analysis by halting every other process and performing its analysis. It is so rigorous, that it halts even if the users does a simple job, like moving the mouse. It is of the type Real Program (mentioned in the literature survey - Types of Benchmarking). The output of this software is a set of scores for each program it has run and the final score. These scores have been determined by the standards

of the Asus Company.

The average scores will be tallied, and the other parameters will be measured.

4.2.2 Attack Phase

- In the attack phase, we will be using a set of 7 basic attacks, namely
  - Phishing
  - Spear Phishing
  - Backdoor (Reverse TCP + Random Multimedia Files Execution)
  - Backdoor (Execution Of Buffer Overflow Virus)
  - EvilGrade
  - Infections Media Generator
  - Make an Executable Virus
- The details of these strategies are mentioned in the literature survey and also in the reference section. For each attack, the basic parameters, as explained in Phase 1, will again be calculated, and an additional parameter set called Type Of Compromise will be filled. Type Of Compromise is a field in which we will be mentioning the sort of compromise that the attacker is able to achieve, whether its just OS details, or access of files, etc.
- The software we will be using will be the same, HWMonitor and RealBench Benchmarking tool.
- Another field, which will be filled, is called Chain Of Custody form. [23]. Chain Of Custody form is a form which is filled when a case is reported and needs to be documented. This form is powerful enough to be presented in a court room. This form will be filled to just document and describe the attack which was experienced by the client system (Windows 10 Pro).The average scores will be talied, and the other parameters will be measured.
- Using the information derived at Phase 2 from the parameters calculated and the description on the Chain Of Custody form, we will form an Attack report. The attack report will consist of the steps taken to perform the attack, the time required for the compromise and the level of compromise. The

information of the ports which were accessible (which is sometimes not disclosed), and the information which was derived will also be documented.

4.2.3 Defence Phase

- The defensive strategies that we will be using are softwares that already exist in the market/real world, and we won't be devising/making a new strategy.
- The defence phase can be done in two ways:
  1. Enable all defence strategies, perform the attacks and the shut down a certain number of defence strategies so as to come up with a minimal set of defence strategies which will still protect the system, [5], or
  2. To test out each defence strategy against a particular attack and check its effectiveness.
- We will be doing the test of type 2, since the benefit is that we will end up judging each defensive strategy in the most radical way possible. We will be testing 9 different defensive strategies, namely
  - Blacklist and Reputation Systems
  - Intrusion Prevention System
  - Disk Encryption
  - Host Based Firewall
  - OSQuery
  - Intrusion Detection System
  - Web Application Firewalls
  - Automated Response and Remediation
  - Backup And Rollback
- Normally, a defensive strategy is of 3 types, Prevent, Detect and Recover. The project intends to cover up all these three areas. For each defence against a particular attack, the basic parameters, as explained in Phase 1, will again be calculated, and an additional parameter set called Compromise Reduction will be filled. Compromise Reduction is a field in which

cross checking of the Type Of Compromise field of the attack strategy and a check upto what extent does the defensive strategy nullify these compromises will be done.

- The software we will be using will be the same, HWMonitor and RealBench Benchmarking tool. The average scores will be tallied, and the other parameters will be measured.
- At the end of the defensive strategy, with respect to what data is obtained, the approach is to shortlist/qualify a set of such defensive strategies which will successfully immunise the client operating system (Windows 10 Pro) against these basic attacks performed. A full report with respect to each of the defensive strategy used, and the reduction of compromise, will be written and a set, with justification of the set of defensive strategies will also be put forth, which is the end goal of the project.

4.2.4 Implementation of API

After the second and the third phase, it will be evident that some features among the above would be very much useful to provide minimal security against the given attacks. These features will be programmed with necessary improvisations as packages, which hence can be accessible via another software, or any other means.

Chapter 5

Implementation

5.1 Backup Module

- When user clicks back up he gets the following options : Normal Backup,Back up with encryption and Back up with compression.
-
  - Under Normal Backup we have two types of back up
    - File back up
    - Directory Back up
  - Under Back up with encryption
    - The directory chosen gets encrypted using AES encryption and backed up.

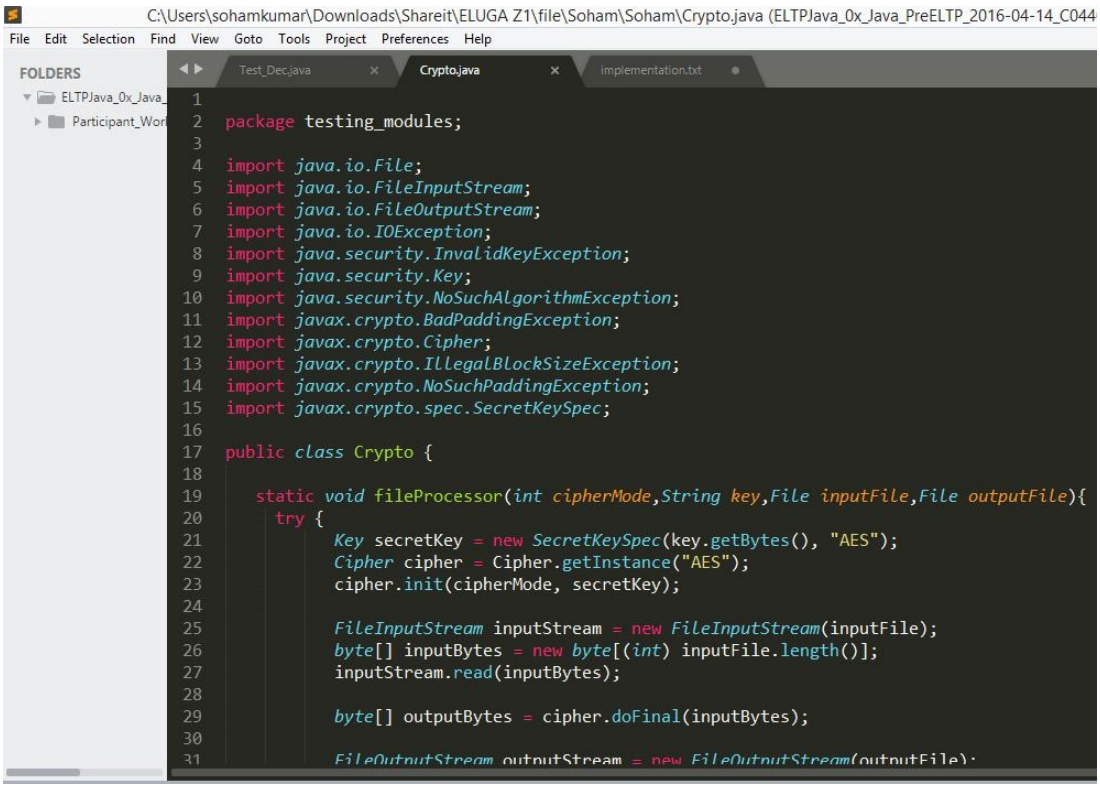

Figure 5.1: (a) Encryption code

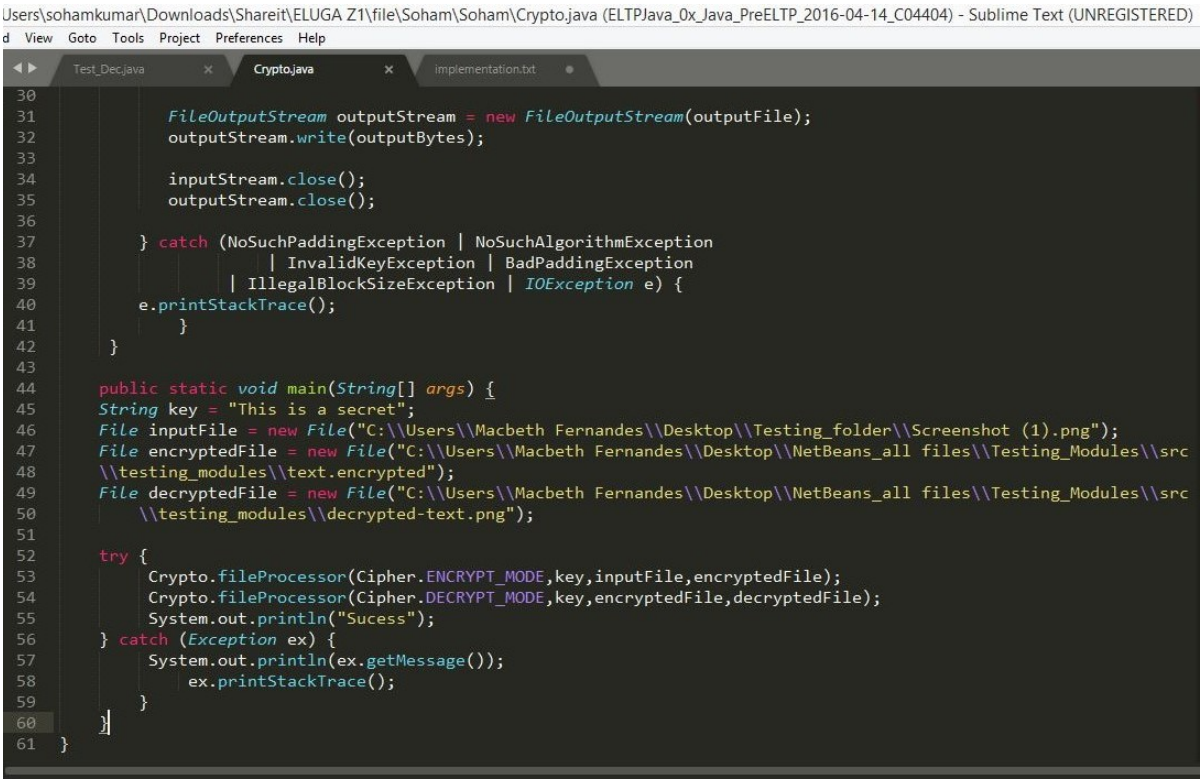

Figure 5.2: (b) Encryption code

– Under Back up with compression
    * The directory chosen gets compressed by 7zip and backed up.

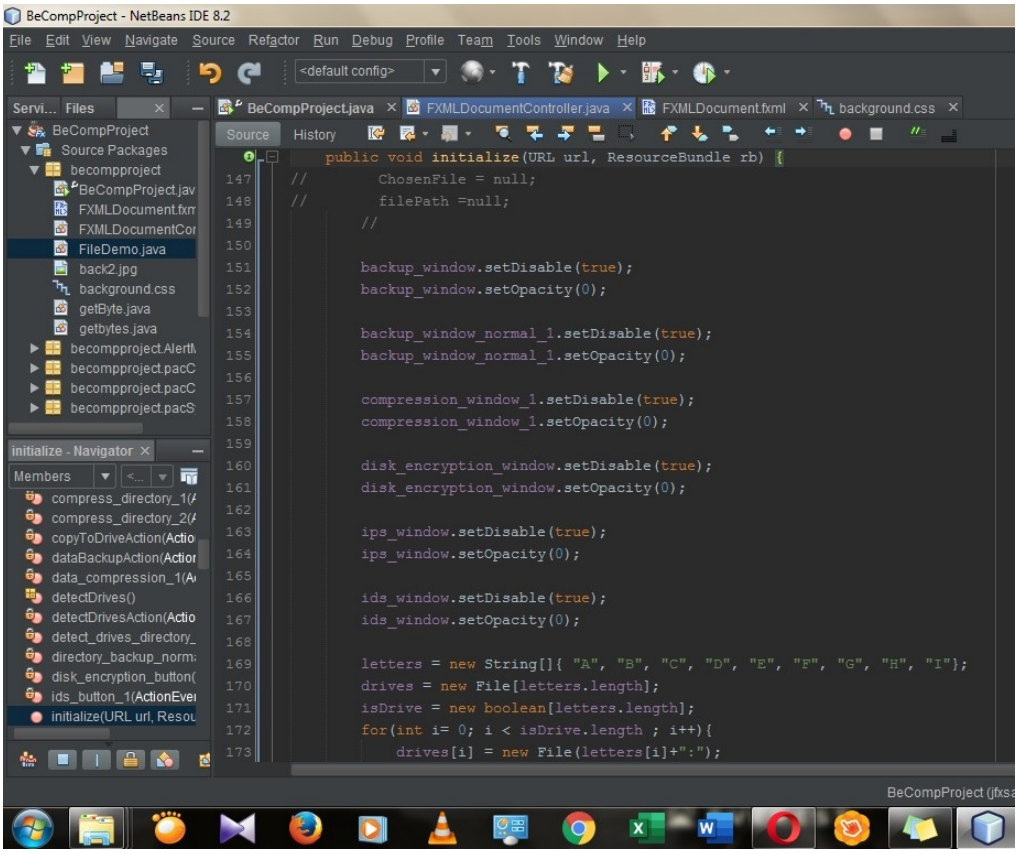

Figure 5.3: (a) Backup code

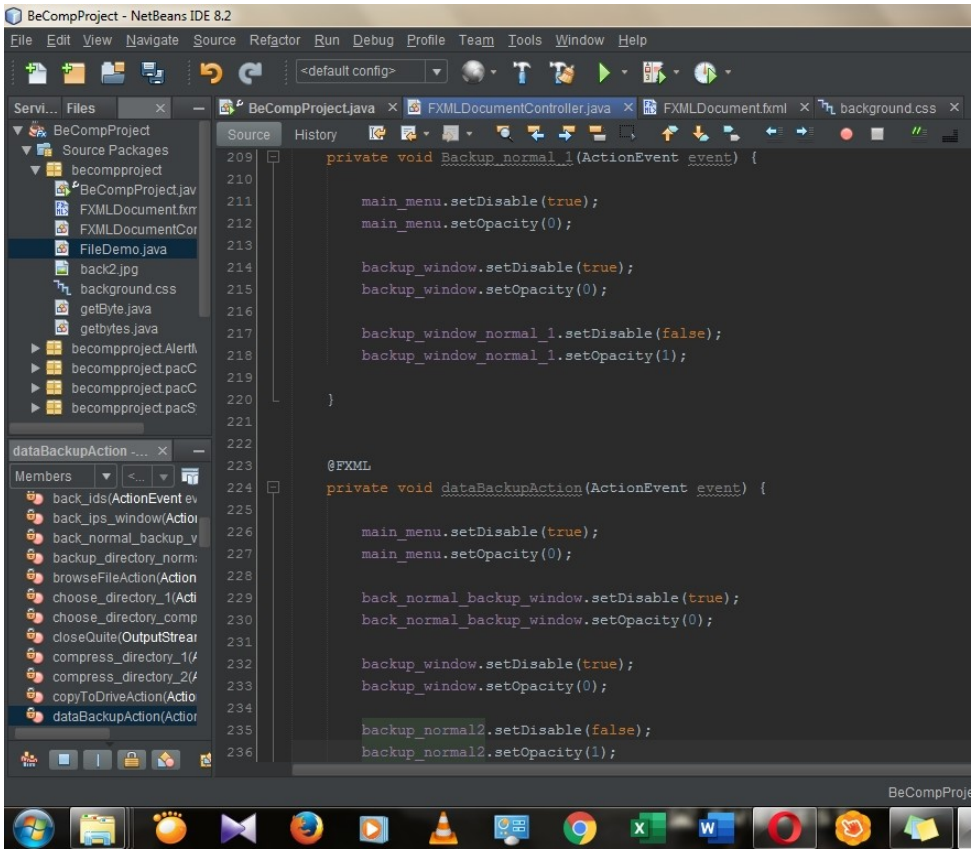

Figure 5.4: (b) Backup code

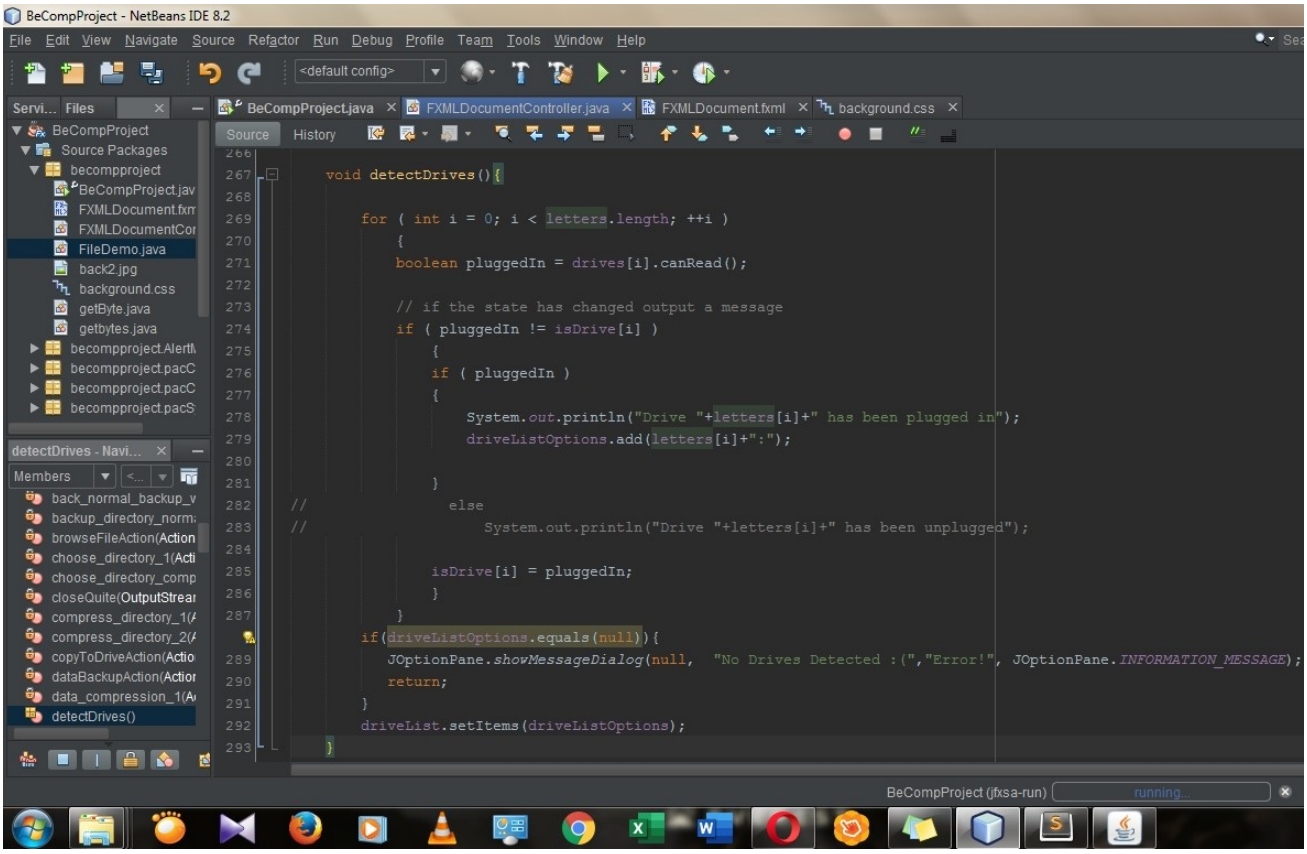

Figure 5.5: (c) Backup code

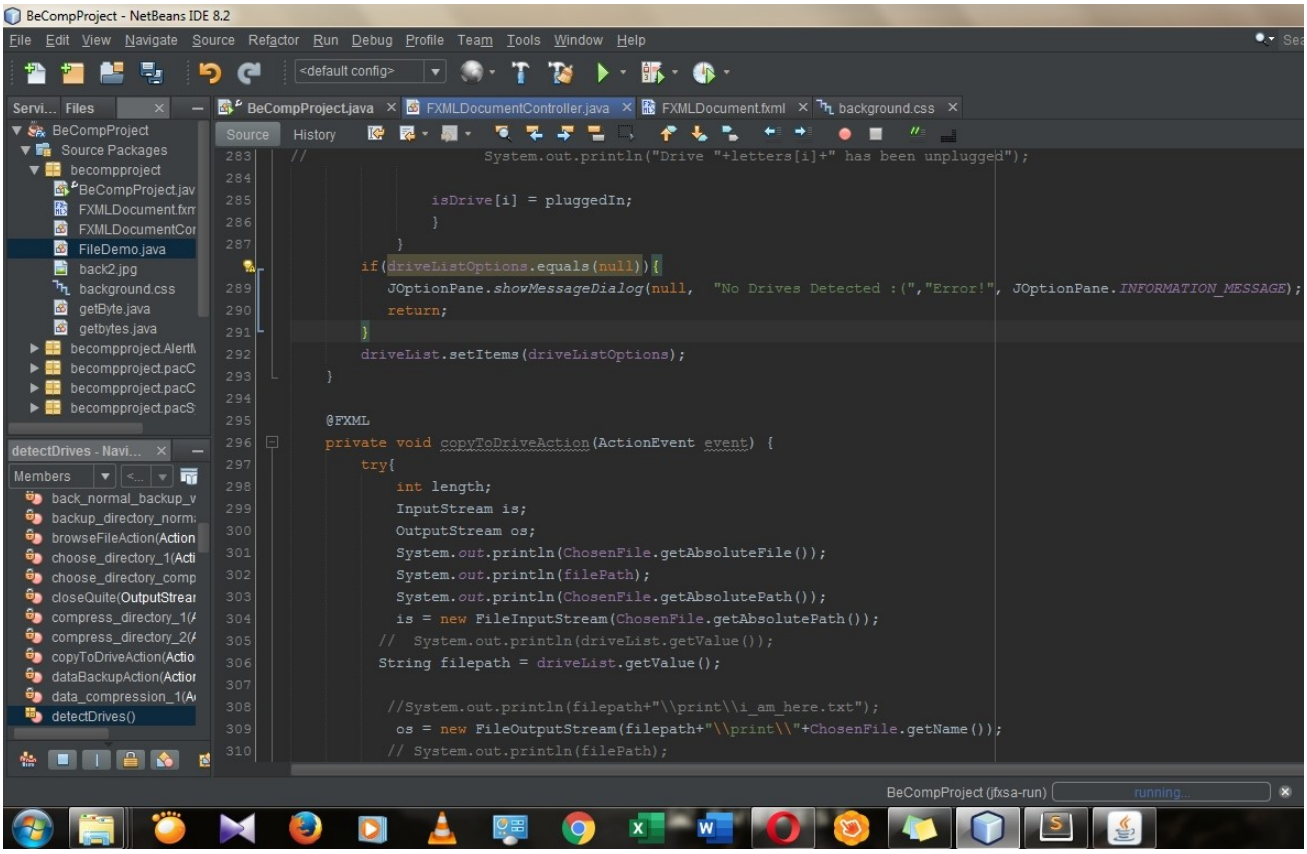

Figure 5.6: (d) Backup code

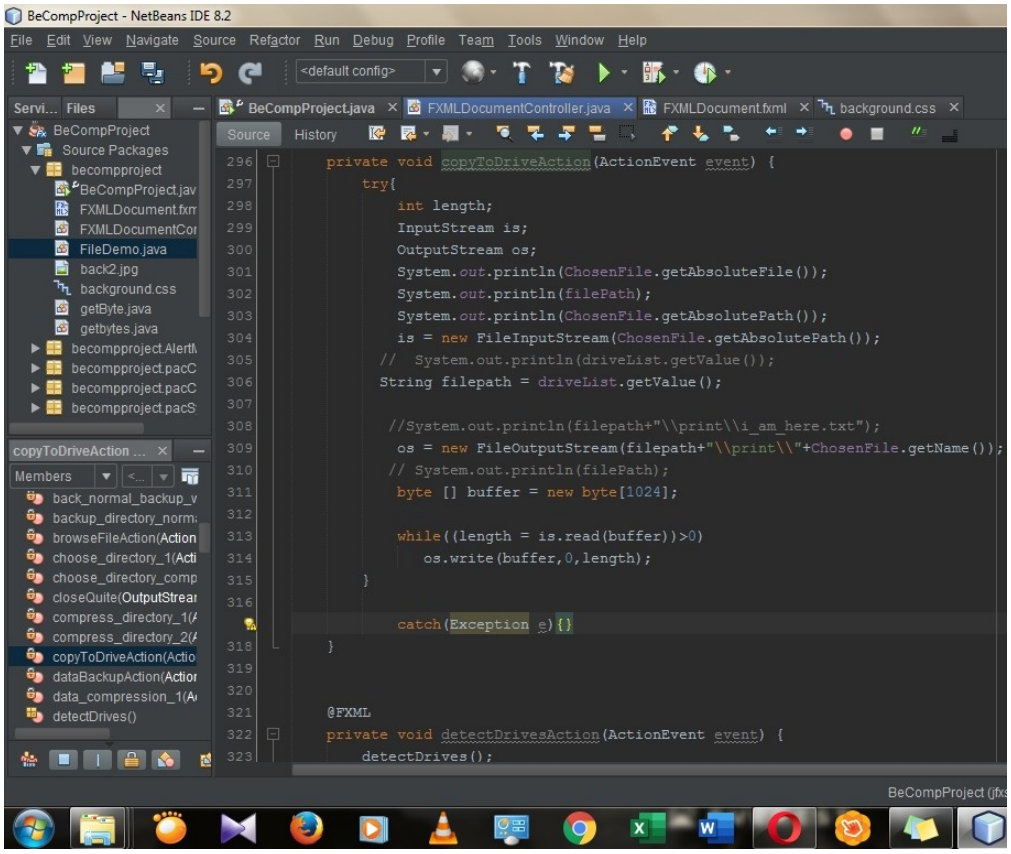

Figure 5.7: (e) Backup code

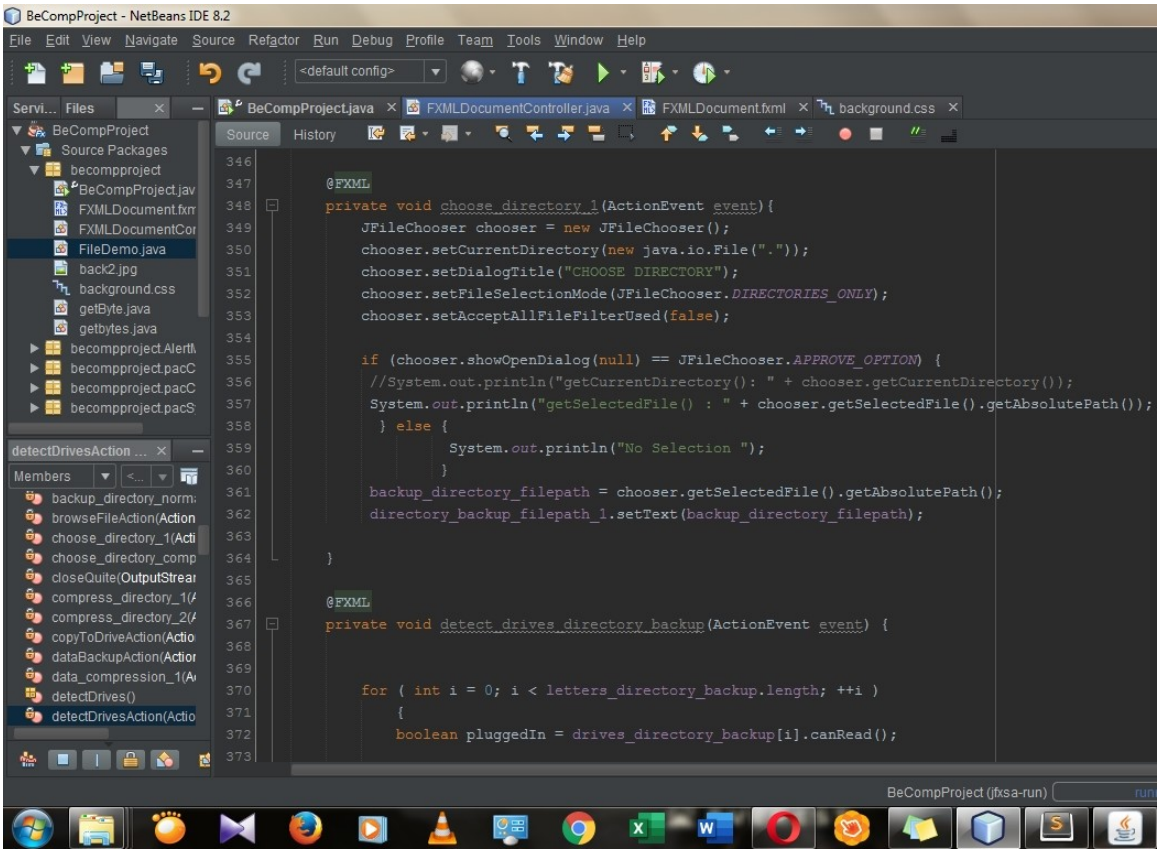

Figure 5.8: (f) Backup code

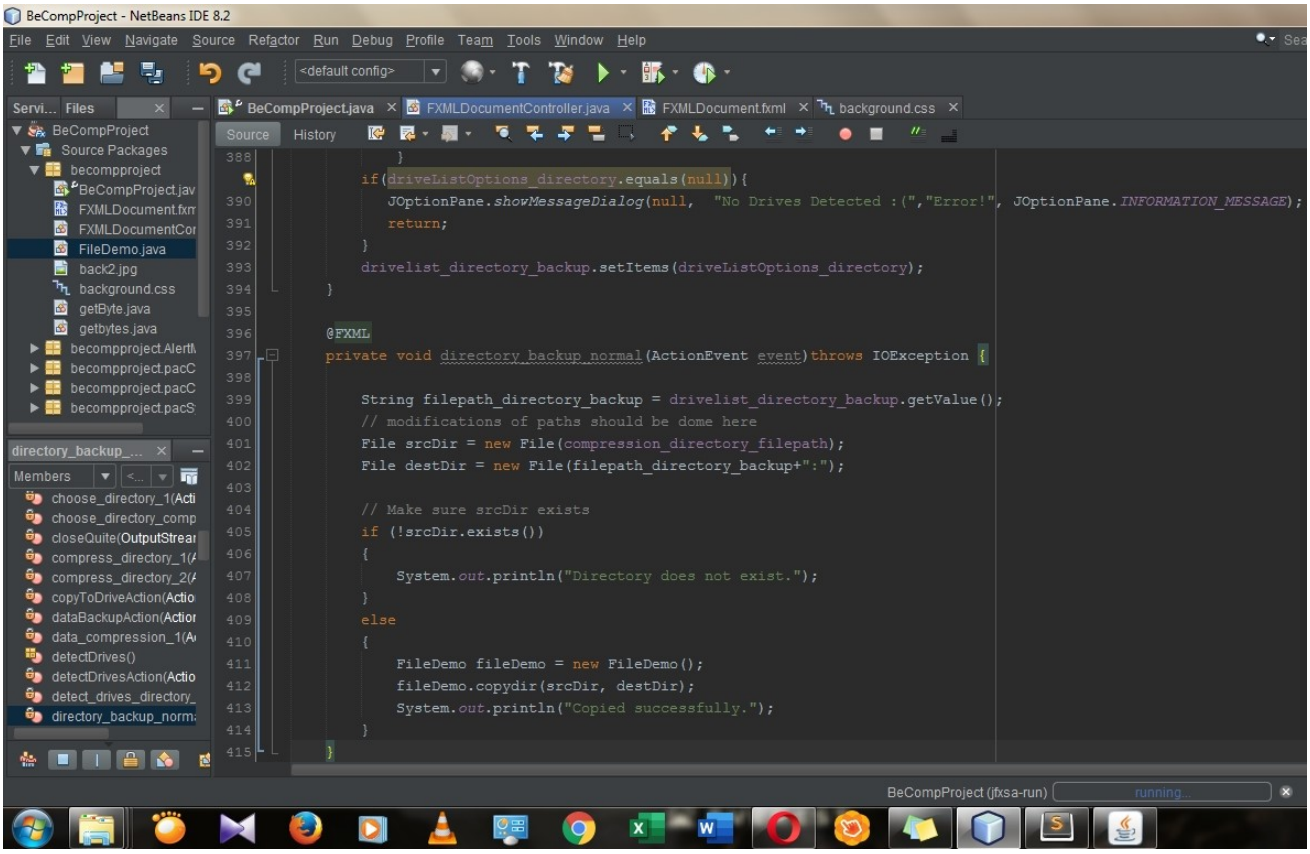

Figure 5.9: (g) Backup code

5.2 Compression Module

1. Choose the directory for compression.
2. Under the Directory list all files in it.
3. All files listed get compressed using 7zip utility package.
4. The compressed directory is then saved at the directory location.

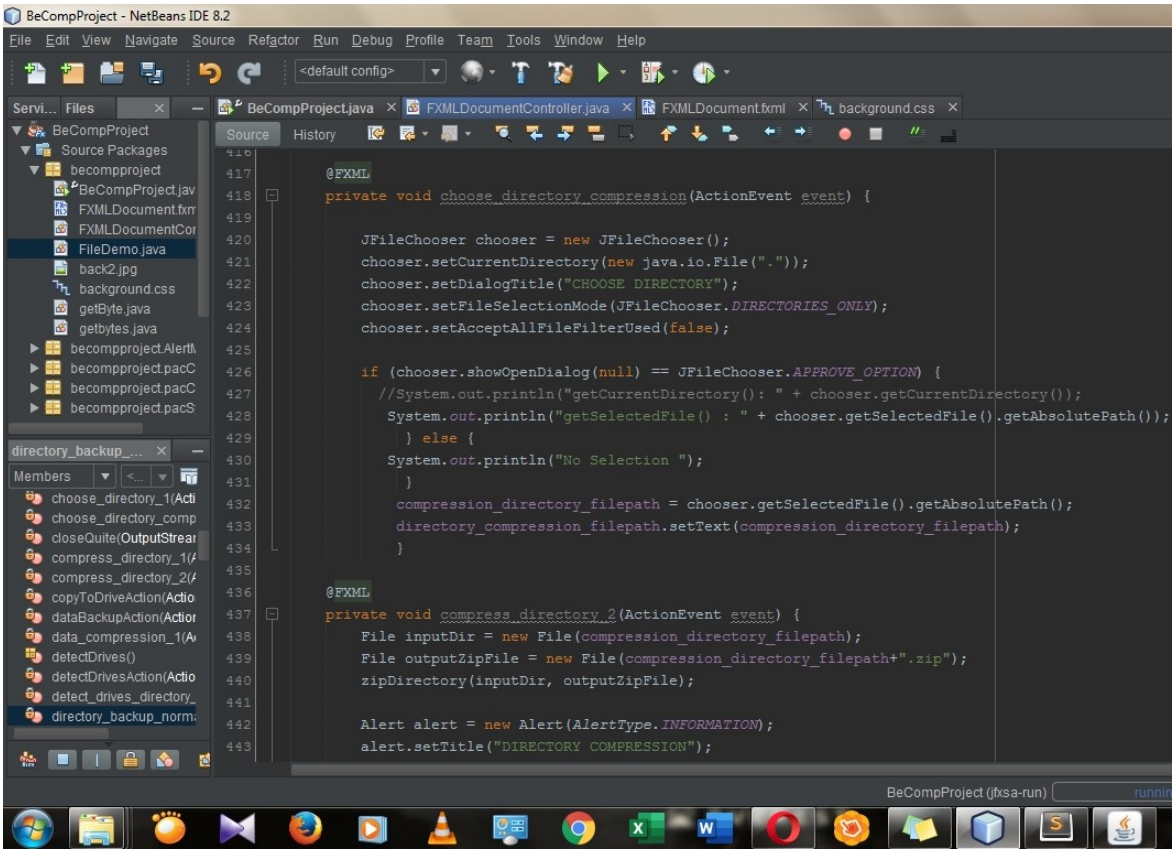

Figure 5.10: (a) Compression code

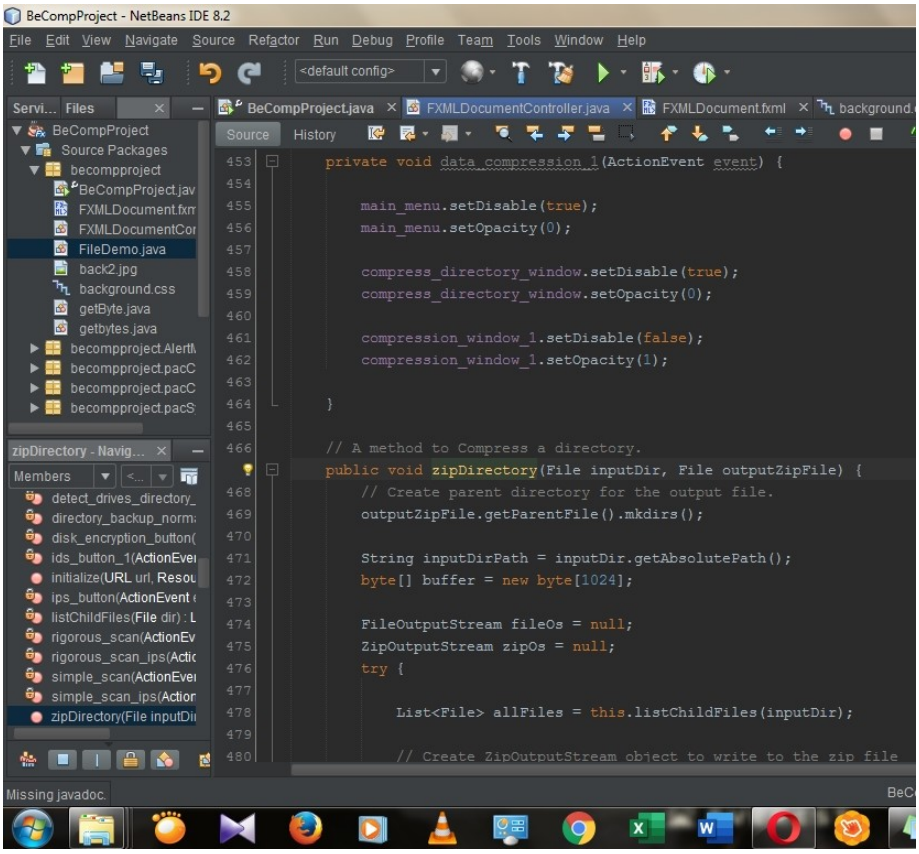

Figure 5.11: (b) Compression code

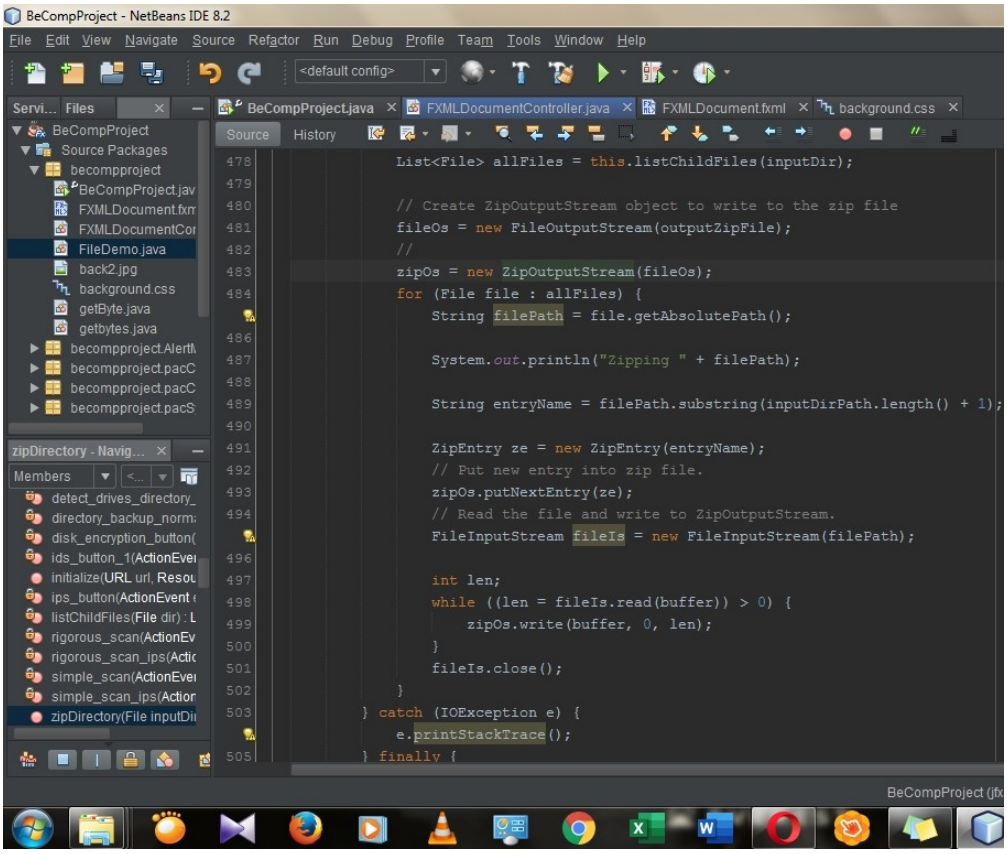

Figure 5.12: (c) Compression code

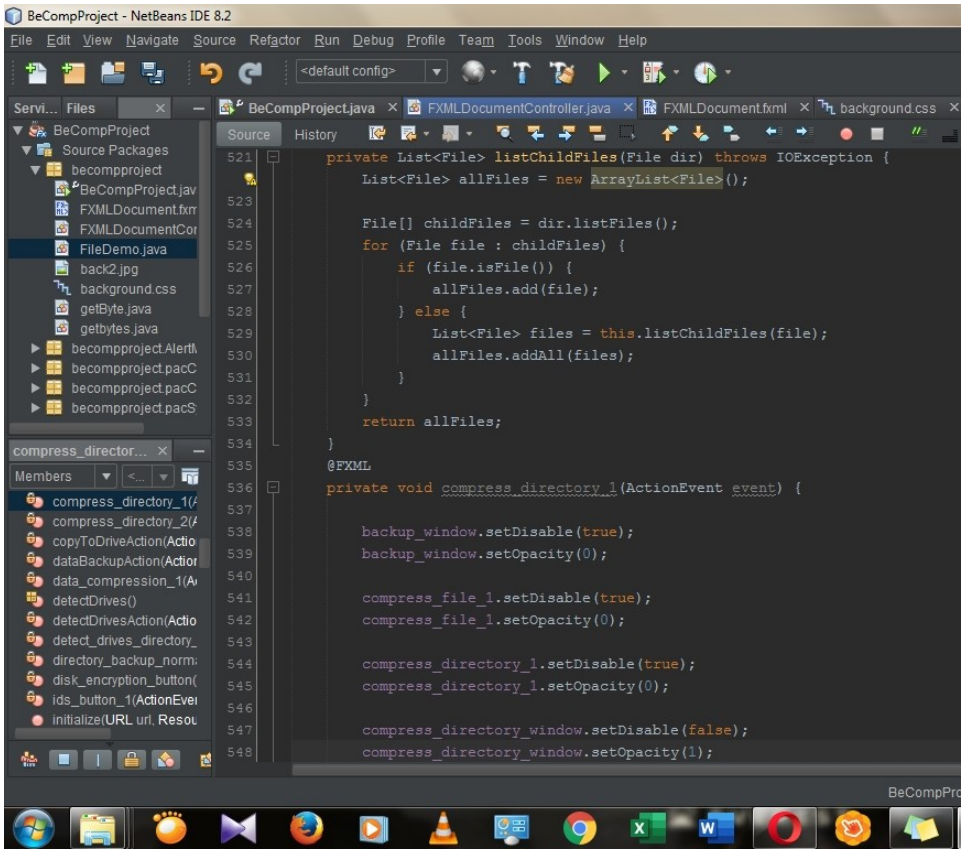

Figure 5.13: (d) Compression code

5.3 IDS

It provides the user with 2 scans i.e Simple scan and Rigorous scan.

5.3.1 SIMPLE SCAN

1. The first step would be to initiate a packet capture. The packet capture captures packets and stores it as a pcap file.
2. pcap file is used as input for analysis of calculating three packet parameters which are data packet capture average packet rate and average packets.
3. Initialization of threshold value for the three packet parameters.

4. If the parameters exceed the threshold value then it indicates that the session in which the packets were recieved was an abnormal session and system could be at risk.

5.3.2 RIGOROUS SCAN

Point 1) and 2) is same as SIMPLE SCAN

3. 1) and 2) is repeated 10 times along with their corresponding parameter values.
4. Standard deviation is calculated for the the entire set of values derived from the 10 scans.
5. Standard deviation is compared with your threshold values of every scan.
6. 4) of Simple scan.

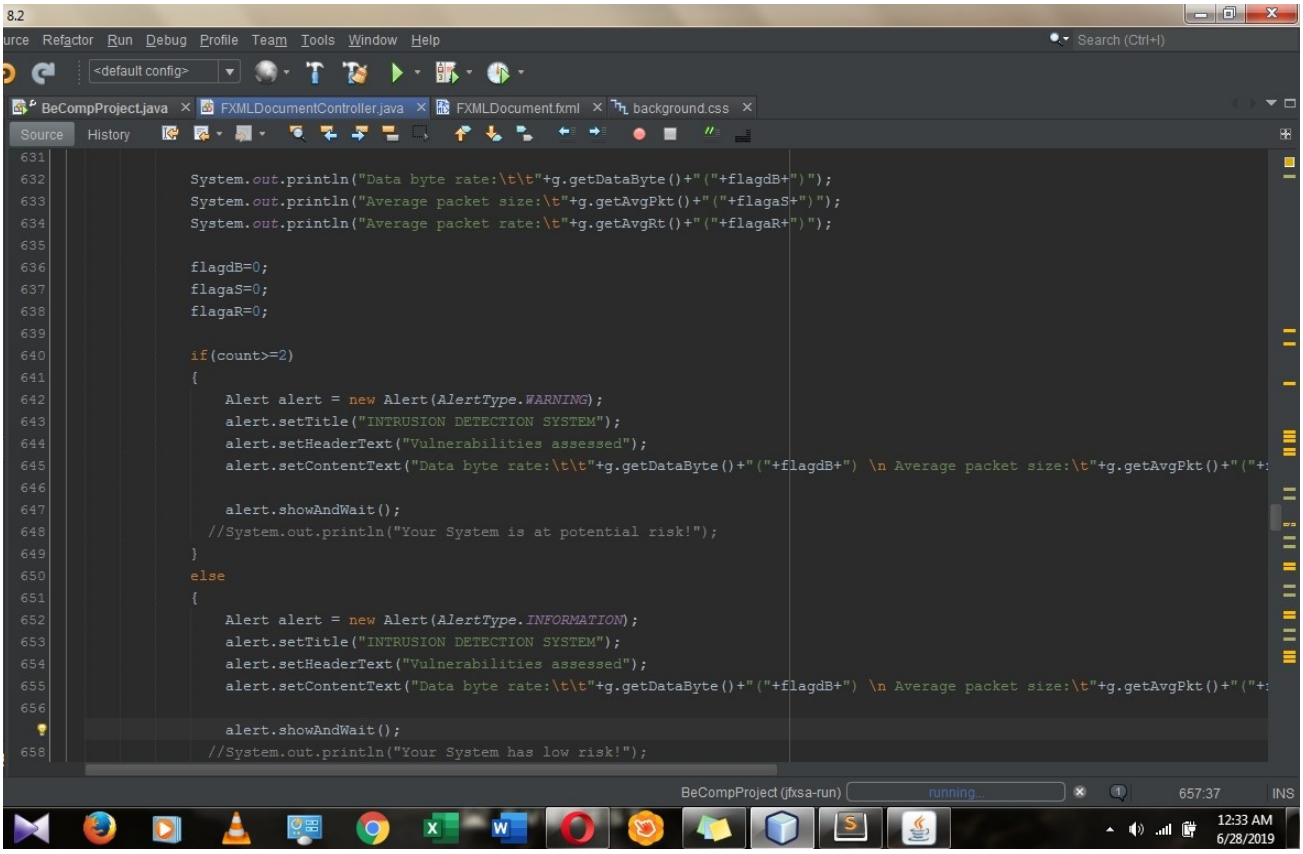

Figure 5.14: (a) IDS code

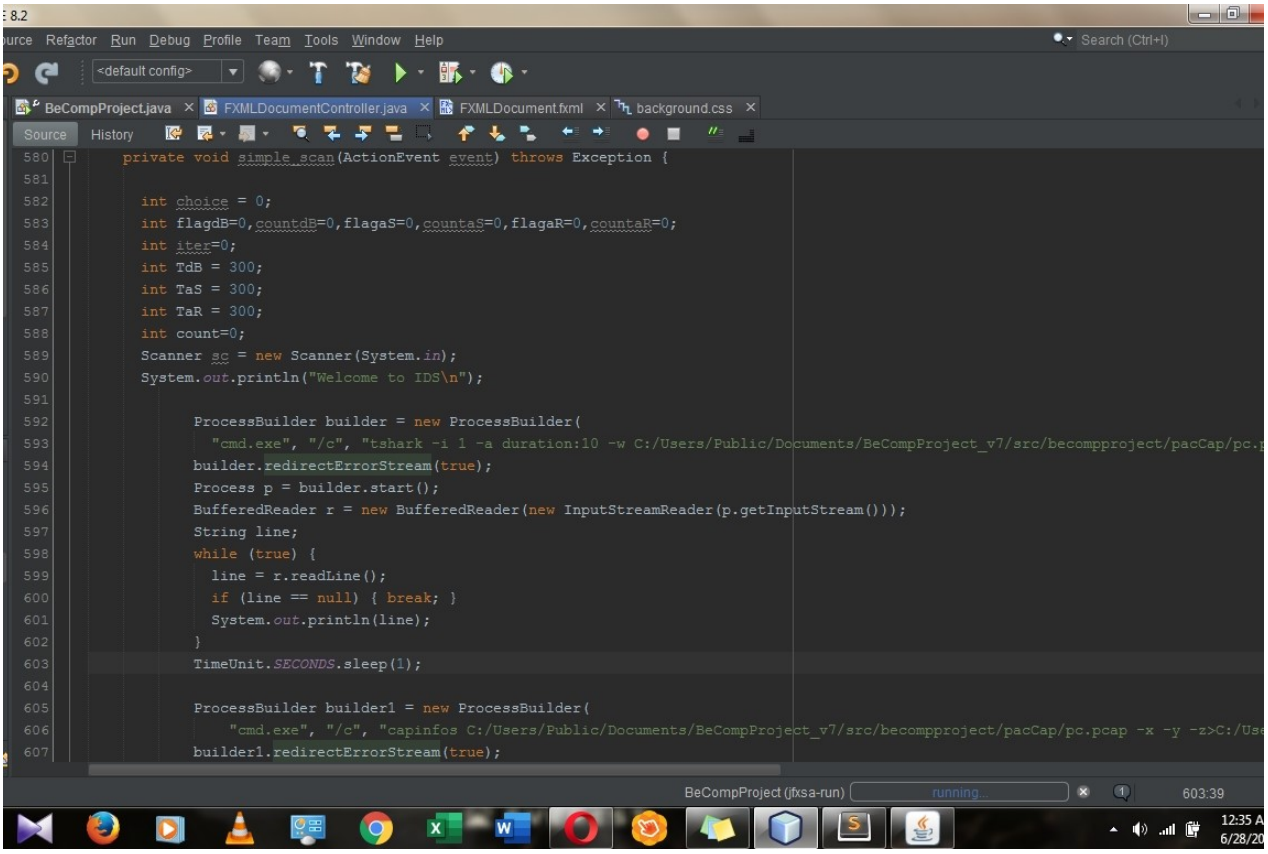

Figure 5.15: (b) IDS code

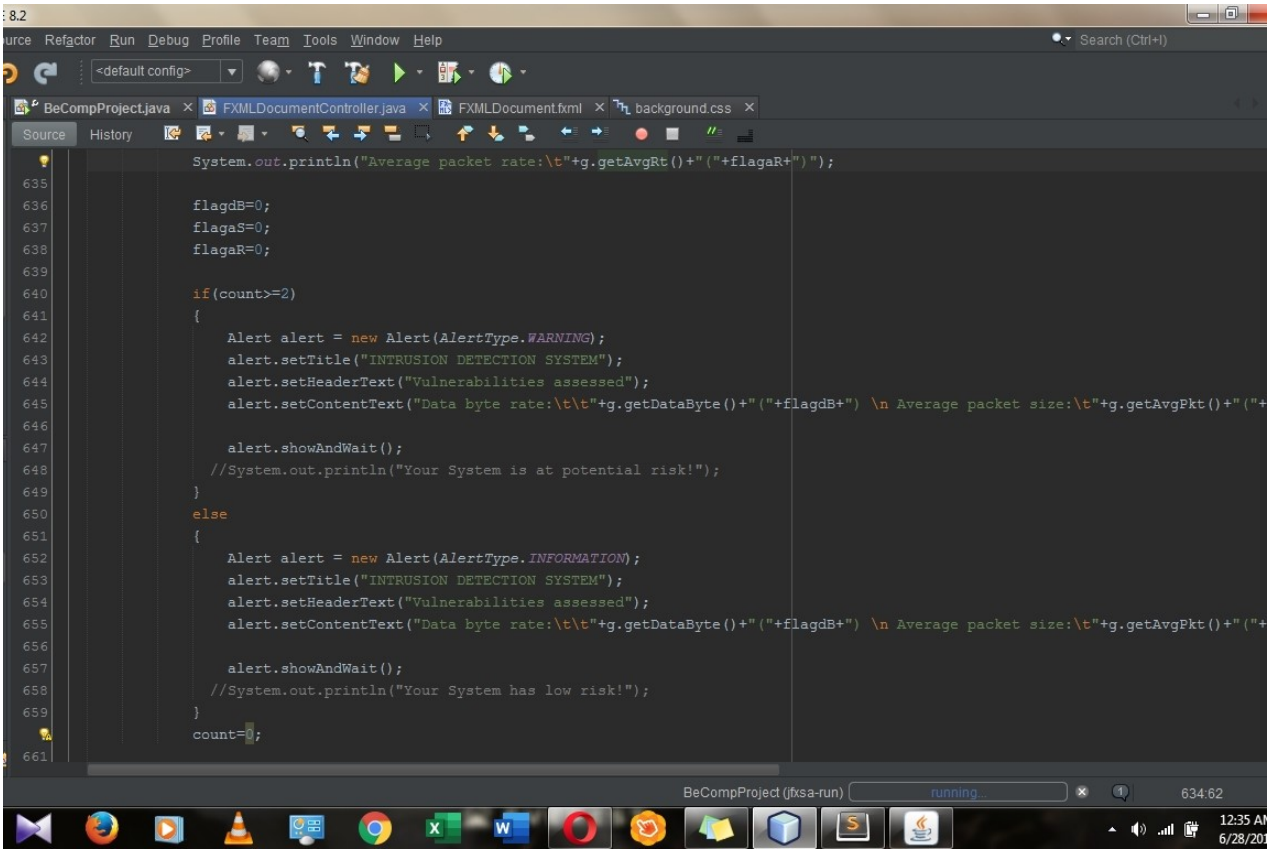

Figure 5.16: (c) IDS code

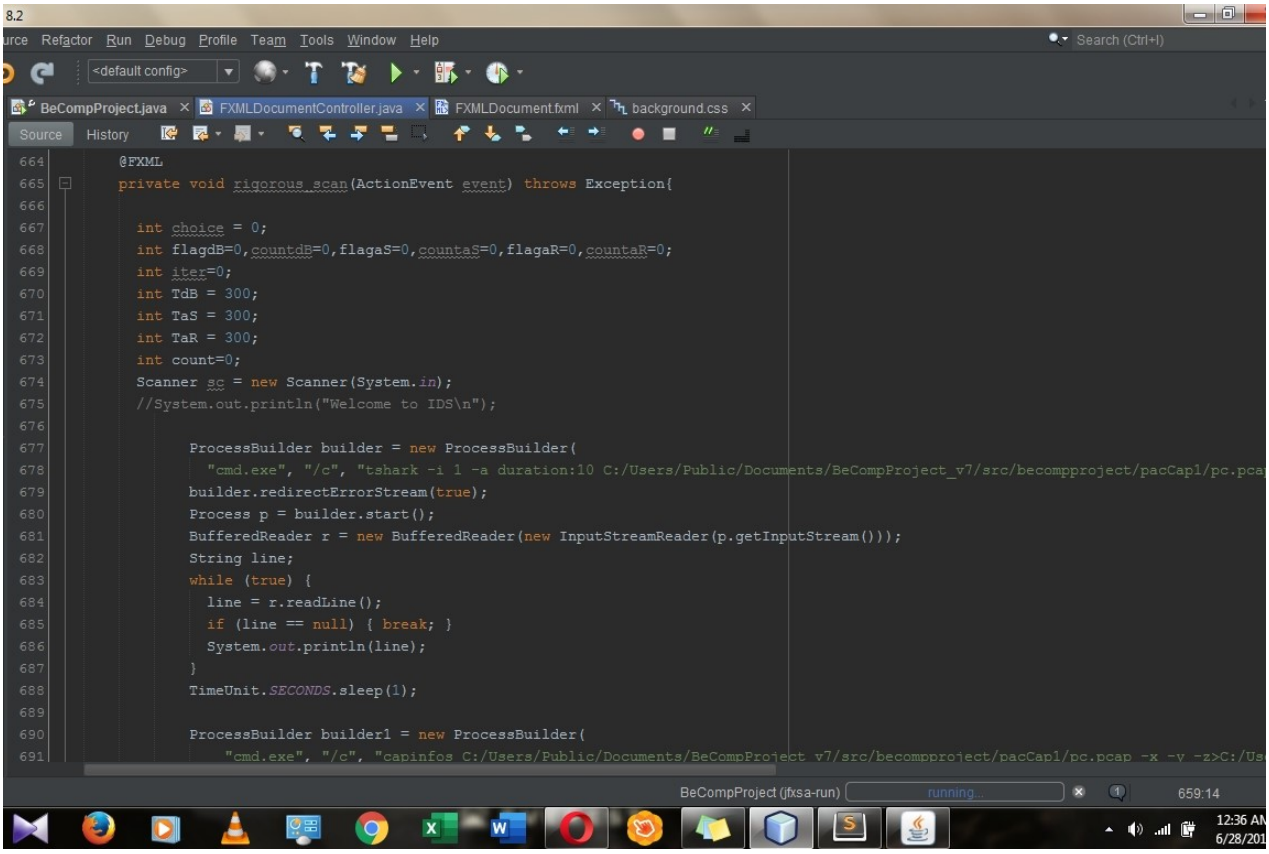

Figure 5.17: (d) IDS code

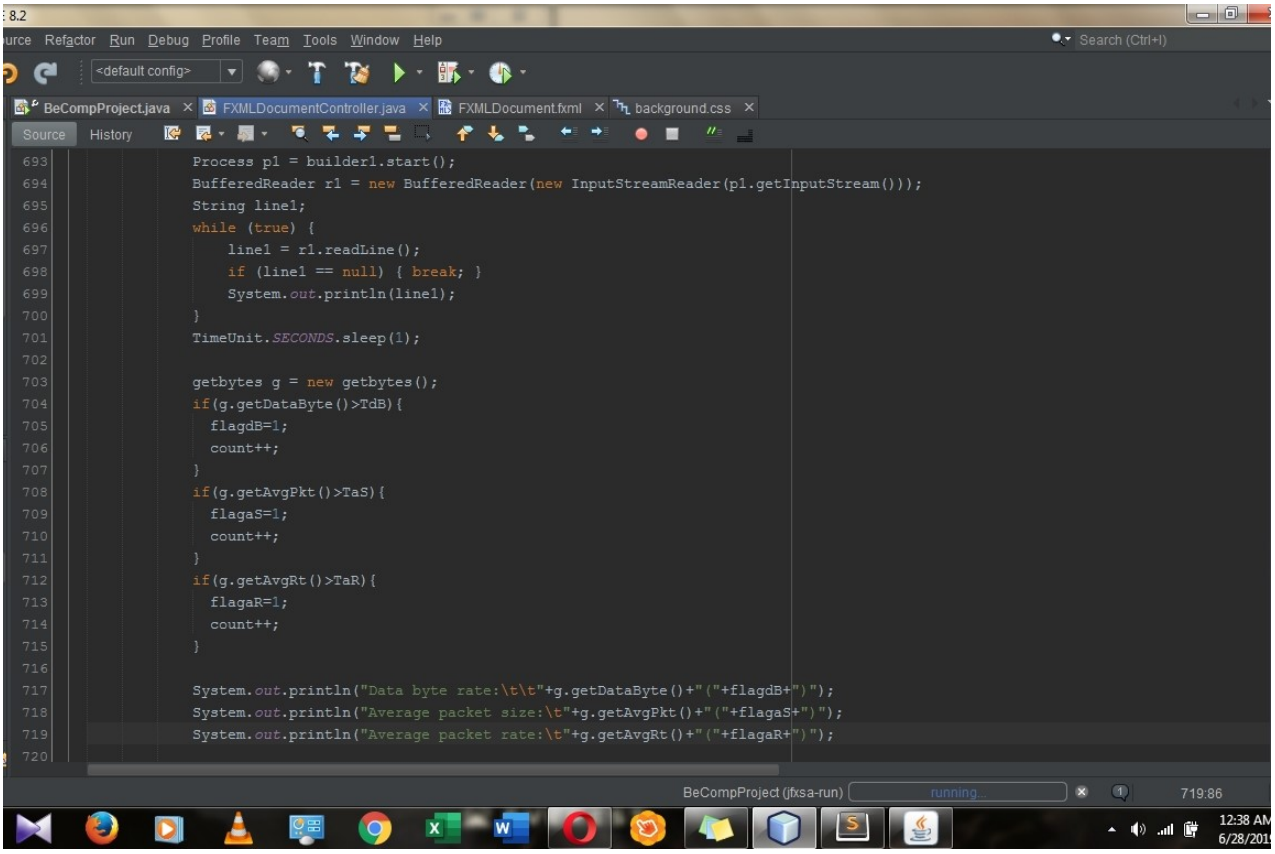

Figure 5.18: (e) IDS code

5.4 IPS

It provides the user with 2 scans i.e Simple scan and Rigorous scan.

5.4.1 SIMPLE SCAN

1. Same as SIMPLE SCAN of IDS
2. If system is at risk then we drop a set of packets for 10 seconds and the network connection is resetted.

5.4.2 RIGOROUS SCAN

1. Same as Rigorous scan for IDS
2. Same as IPS 2) under SIMPLE SCAN.

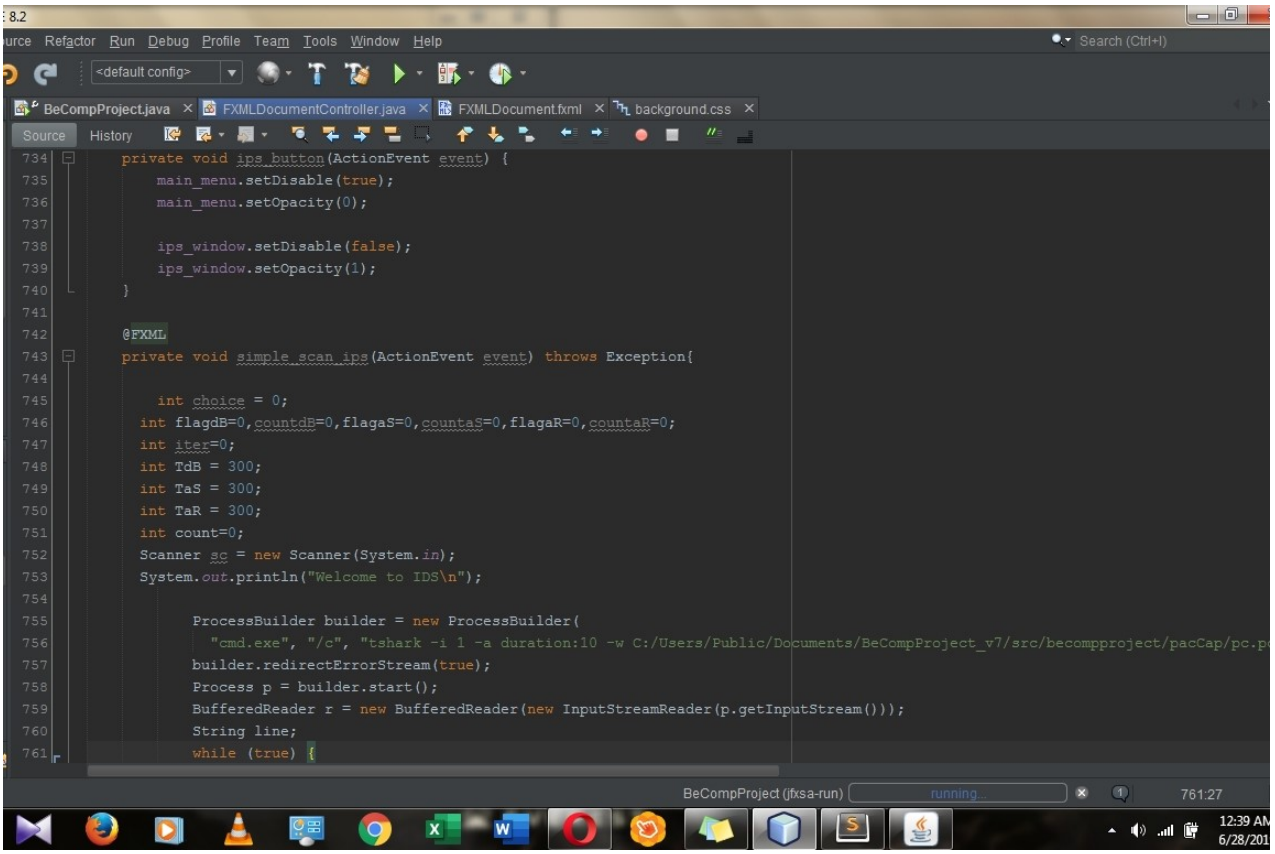

Figure 5.19: (a) IPS code

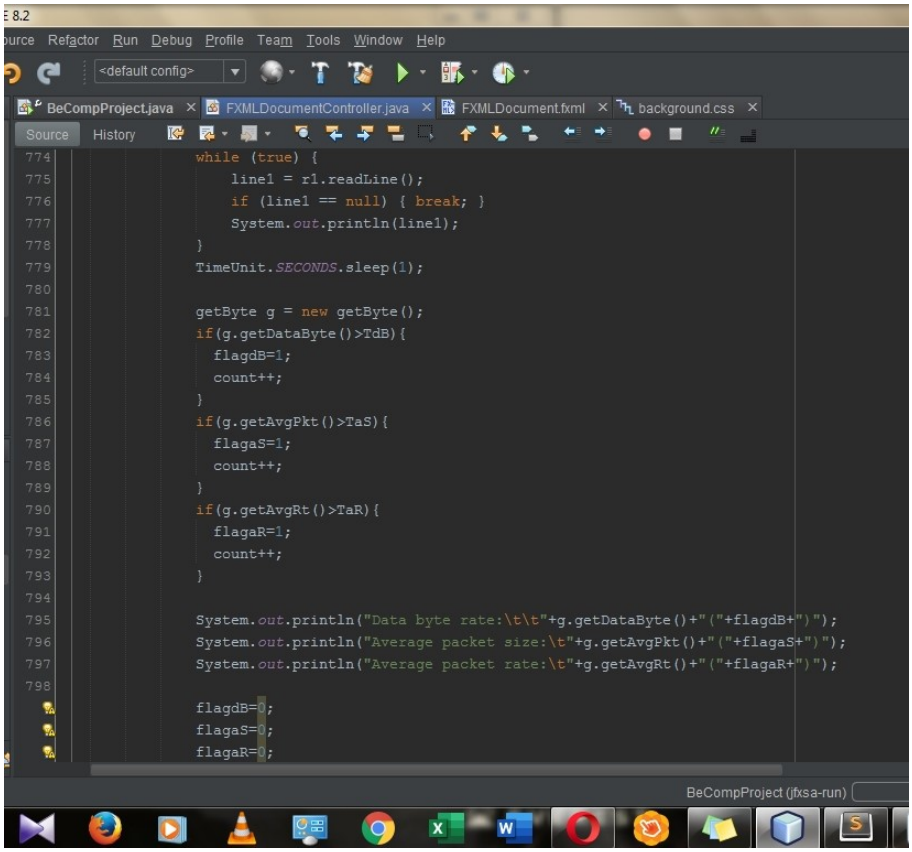

Figure 5.20: (b) IPS code

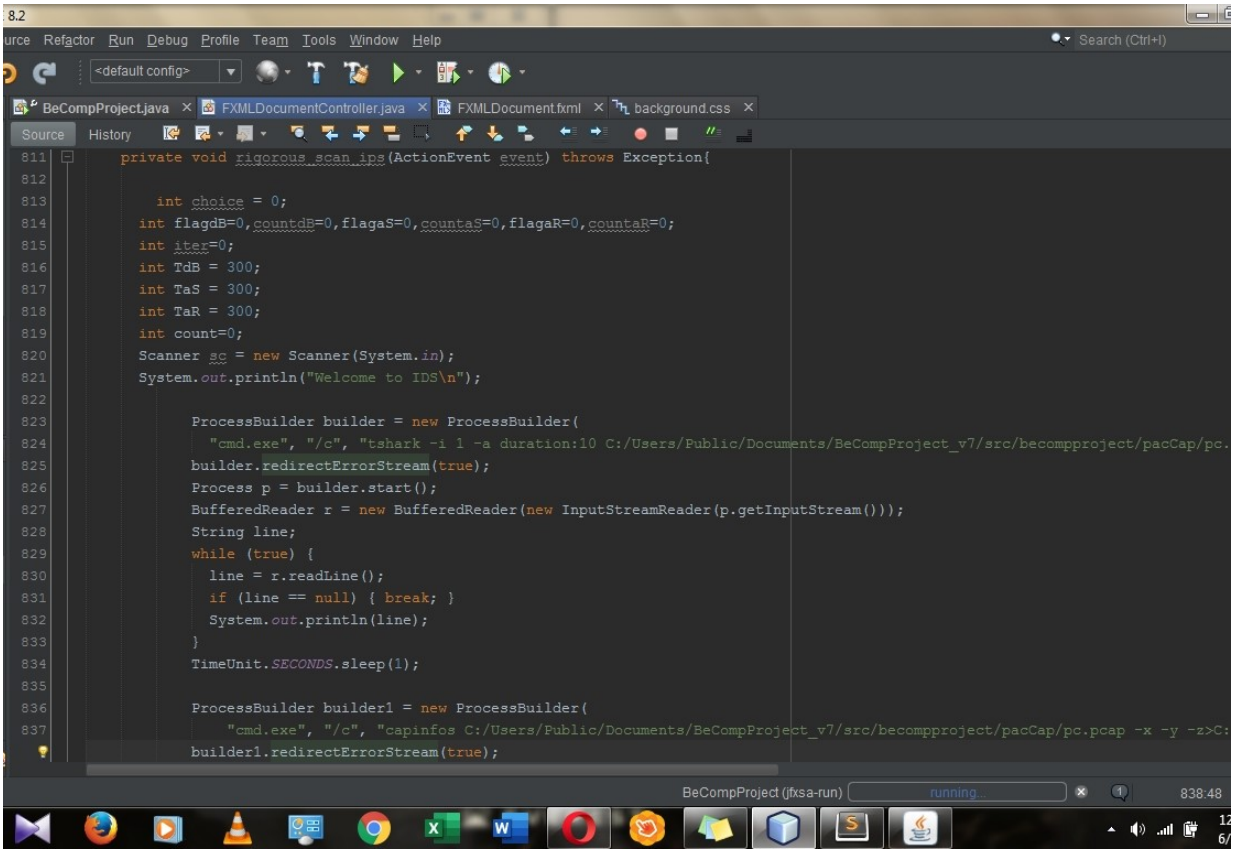

Figure 5.21: (c) IPS code

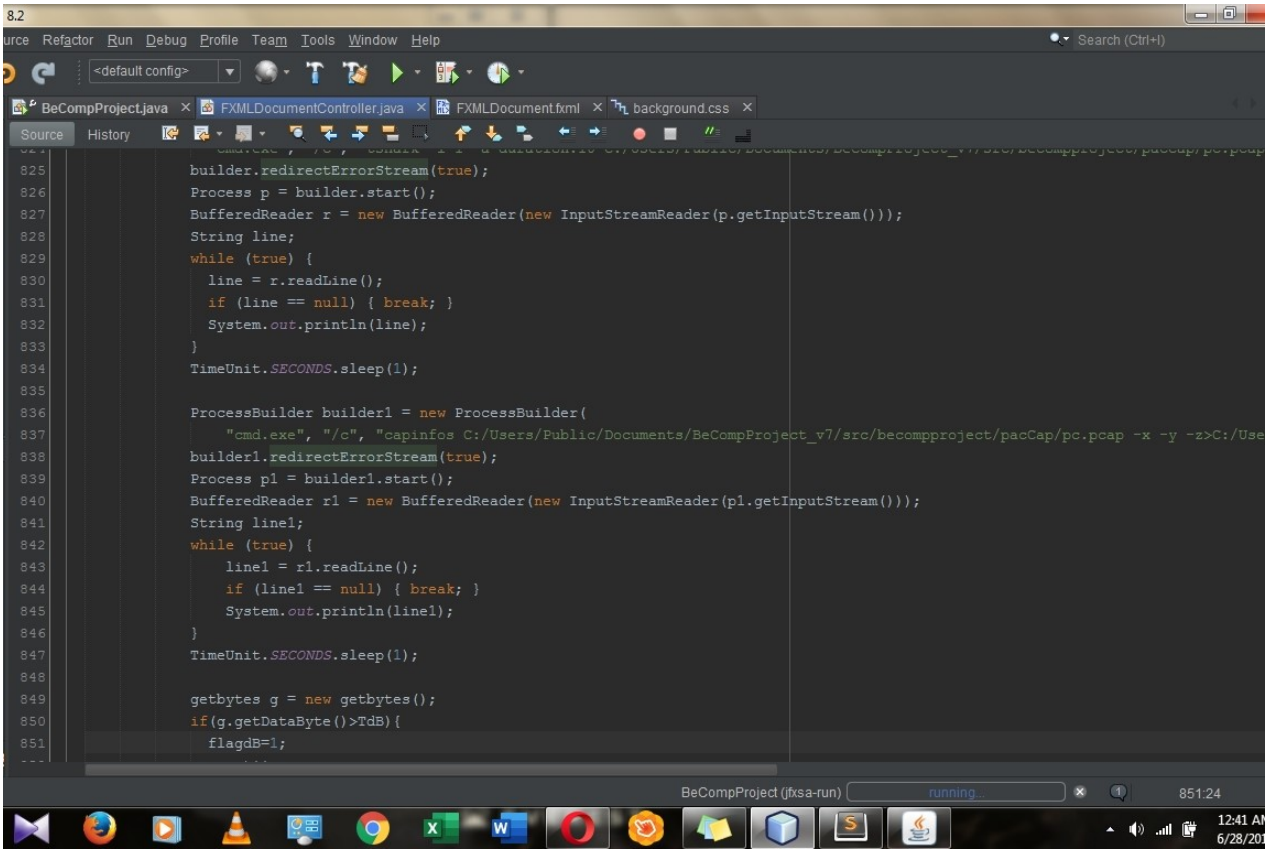

Figure 5.22: (d) IPS code

Chapter 6

Experiments Conducted

A1	Phishing	P1	Temperature (in deg C) vs Time (in seconds): Core #0
A2	Spear Phishing	P2	Clock Speed (in MHz) vs Time (in seconds): Core #0
A3	Backdoor (Reverse TCP + Random Multimedia Files Execution)	P3	CPU Utilization (in %) vs Time (in seconds): Core #0
A4	Backdoor (Execution Of Buffer Overflow Virus)	P4	GT Temperature (in deg C) vs Time (in seconds): Core #0
A5	EvilGrade	P5	HD Graphics GPU Utilization (in %) vs Time (in seconds): Core #0
A6	Infections Media Generator	P6	HD Graphics Clockspeed (in %) vs Time (in seconds): Core #0

D1	Blacklist and Reputation Systems
D2	Intrusion Prevention System
D3	Disk Encryption
D4	Host Based Firewall
D5	OSQuery
D6	Intrusion Detection System
D7	Web Application Firewalls
D8	Automated Response and Remediation
D9	Backup And Rollback

6.1 Phase 1

The system configurations of the client system is as follows:

- Windows Edition : Windows 10 Pro System
- Processor : Intel(R) Pentium(R) CPU N3540 @2.16GHz
- Installed Memory (RAM) : 4.00 GB (3.89 GB usable)

- System Type : 64-bit Operating System, x64-based processor.

The above information is retrieved from ≫ Control Panel \System and Security \System. The procedure followed in this phase is explained in the Objectives and Methodology Phase 1. We have used a benchmarking tool, HWMonitor.

Figure 6.1: Phase 1: GT Temperature (in Deg C) vs Time (in seconds) :Core #0

6.2 Phase 2

6.2.1 Attack 1: Phishing

Software used: NMAP and ZENMAP
Steps:

- NMAP
  1. Open Windows and Kali Systems.
  2. Make sure both have internet access.
  3. Check IP and gateway of both.
  4. Open Nmap
     (a) Read Information Commands,
     (b) To check your gateway IP (KALI system) and which all computers are there in it. #route -n
     (c) To gather IP address: Nmap -sP 192.168.236.2 -254
- ZENMAP
  1. Type the IP address at Target section.
  2. You have scan options:
     – Don't try intense scan for basic information like which OS.
     – Try slow, comprehensive scan or regular scan (takes times).

  – INTENSE SCAN:
    1. Get to know open ports.
    2. Won't get detection, can take some time.
    3. Does OS detection, can take some time.
    4. Shows filtered ports (firewalls).
       (a) To check out MAC address, which company, etc.
       (b) Check MAC Address
           Copy MAC address from Zenmap paste it there and check.
    5. You also get uptime and network distance (min 1 hop)
    6. Router to router hop.
    7. Can check topology.
    8. Shows TCP Sequence Prediction:
       If Difficulty: 263? Sequence Prediction is easy.
  – INTENSE SCAN + UDP
    UDP scan shows us from which port we can go ahead and transfer files like pictures, songs with viruses embedded inside them, trojans, etc.
  – INTENSE SCAN + TCP
    1. This is Noisy.
    2. Client can know that you are lurking.
    3. Check if ip address is incremental (130,131,132,...)
  – PING-SCAN
    Just to know if OS is alive.
  – REGULAR SCAN
    Doesn't check everything, so gives wrong information sometimes.

–osscan guess: exact OS (If the client has IDS installed, he will get to know)

6.2.2 Attack 2: Spear Phishing

Software used: SET (Social Engineering Toolkit)
Steps:

1. Open Terminal, execute setoolkit
2. Choose 1: SOCIAL ENGINEERING ATTACKS
3. Choose 1: SPEAR-PHISHING – Targetted.
4. Select 2: Mass Mailer Attack
5. You will get different types of viruses/payloads
   (a) Select 5: Also puts listener on rootkit machine
6. You will get payloads
   (a) What type of listener you want?
   (b) Name it in a way that the victim is compelled to open it.
   (c) Select 1: Reverse TCP Shell
      i. SET IP address for listener: You IP (2960)
      ii. Port to connect back to normally 443 – default
      iii. Saves as /root/.set/template.pdf
      iv. Choose either 1 or 2
      v. Either choose Send Email or Return
7. Open the file manager, navigate to /root/.set
8. The payload will be present as a pdf fill

6.2.3 Attack 3: Backdoor (Reverse TCP + Random Multimedia Files Execution)

Software used: SETOOLKIT
Steps:

1. Open Terminal, execute setoolkit
2. Choose 4: Create Payload and Listener: Creates a virus, if the victim opens it, the listener begins on the attacker's side, and s/he can access everything from the victim's computer.
3. Choose 4: IP address for the payload (reverse): IP address of your computer. Every time it will go to this IP address, connect back to us.
4. Choose what payload you would like to generate

- Windows Shell Reverse TCP
  (a) Spawn a command shell on victim and send back to attacker.
  (b) Choose 1: Select one of the below, 'backdoor executable' is typically the best. However, most still get picked by AV. You may need to do additional packing/crypting in order to get around basic AV detection.
    – Shikata-ga-nai
    – No Encoding
    – Multi Encoder
    – Backdoored Executable
  (c) Select 4 for backdoored executable
    – Mention port for listener: 443 or 2960
    – Choose either to start the listener now or later (yes or no)

5. Open file explorer: usr>share>set
   (a) Payload.exe is your payload. Deploy the same on victim PC.
6. Let the user execute this payload.
   (a) On Kali system, execute msf exploit (handler)¿sessions -l
   (b) If the user has executed the payload, sessions on the kali system will have a new entry. Choose the associate number.
   (c) Execute msf exploit (handler)¿sessions -I
   (d) You will now have an access to the victim system

6.2.4 Attack 4: Backdoor (Execution of Buffer Overflow Virus)

Software used: setoolkit, meterpreter
Steps: The first few steps until the backdoored executable payload generation will be exactly the same as the Attack 3 procedure. We will use msfconsole to make this payload persistent and execute once the system turns on again after even a brief shutdown. This is done so that the access is maintained.
Execute:

1. Meterpreter> migrate help
   Here we need the processid, which we will get executing the below command
2. Meterpreter> getpid (say 3880)
3. Meterpreter> ps
   Shows running processes

4. Meterpreter> migrate 3880 (3268)
   3268 would be a processid of an important process in windows, to which this process will be merged with.

6.2.5 Attack 5: Evil Grade

Software Used: Evil Grade
Steps: The methodology here would be simple. We need to simply make a payload under the Evil Grade command. To do this:

1. Run the command: Evil Grade
2. Input the IP address of the listener and the IP address of the server system
3. Then allow to listen for any client trying to connect to server.

6.2.6 Attack 6: Infectious Media Generator

Software Used: SE Toolkit
Steps:

1. Run the command SE Toolkit
2. There will be an option to choose to generate an infectious media (option 5)
3. Upon selecting the option, there will be an option to choose the type of media. We chose an infectious pdf
4. Upon choosing that, we choose the type of payload, we chose reverse TCP. We enter the listener IP and the port to listen to. And the payload is created.

6.2.7 Attack 7: Virus attack

We used a folder bomb virus, which made multiple folders upon execution

The table below provides average values for all the parameters P1 to P6 against the attacks A1 to A7 and Phase 1 scans. Some output graphs are provided as proof of work.
Green Color indicates that the parameters return lower values compared to that of Phase 1. Red Color indicates that the parameters return higher values compared to that of Phase 1.

Table 1: Values obtained in different attacks

	Phase 1	A1	A2	A3	A4	A5	A6	A7
P1	40.86	51.0	52.95	52.885	50.25	53.826	50.25	42.5
P2	2666.0	724.8	523	523	991.6	1395.4	500.5	460.8
P3	5.47	5.335	2.195	0.945	17.52	6.805	3.575	11.29
P4	41.56	51.4	52.885	52.88	41.56	55.252	53.065	55
P5	0.776	0.452	0.46	0.467	0.776	0.786	0.441	0.481
P6	667.98	620	635.3	716	620	644.74	627.2	630.3

Figure 6.2: Attack 4: Temperature (in deg C) vs Time (in seconds): Core #0

Figure 6.3: Attack 3: CPU Utilization (in %) vs Time (in seconds): Core #0

Figure 6.4: Attack 2: HD Graphics GPU Utilization (in %) vs Time (in seconds):Core #0

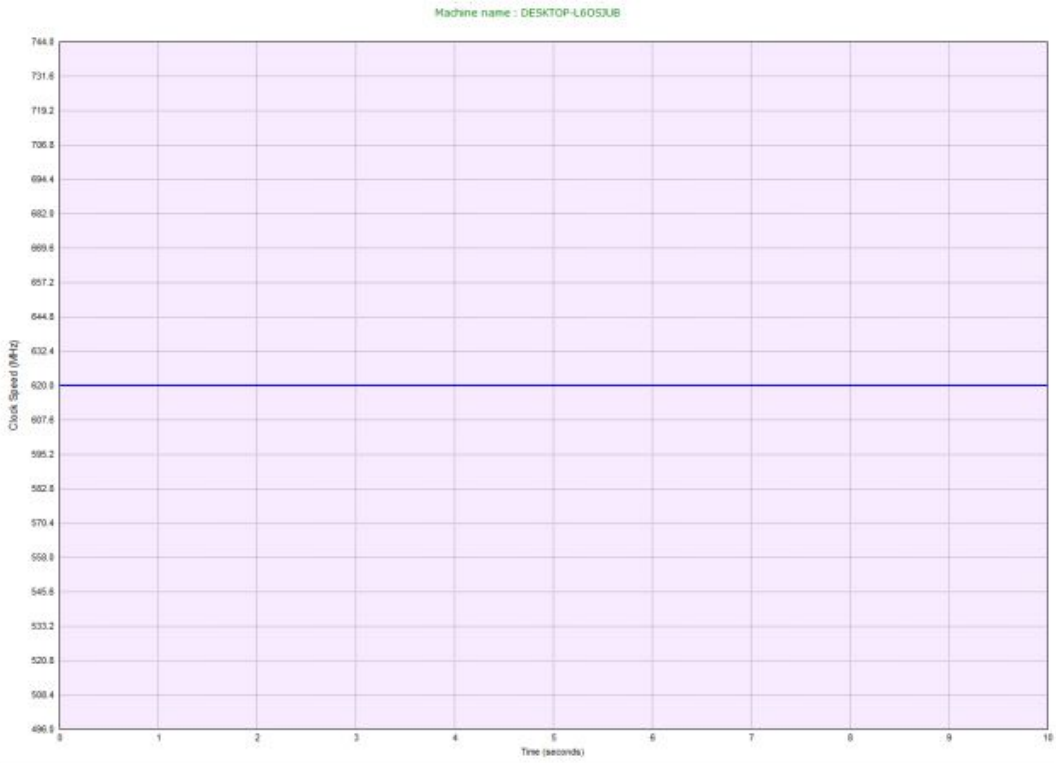

Figure 6.5: Attack 1: HD Graphics Clock speed (in %) vs Time (in seconds)

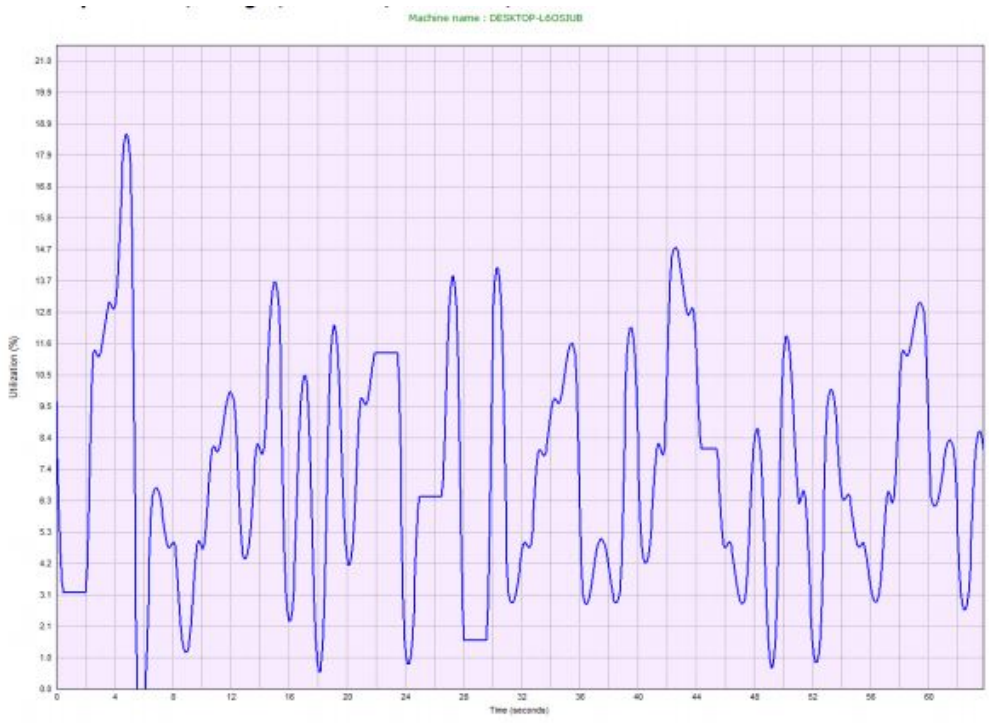

Figure 6.6: Attack 5: Temperature (in deg C) vs Time (in seconds): Core #0

Figure 6.7: Attack 6: Temperature (in deg C) vs Time (in seconds): Core #0

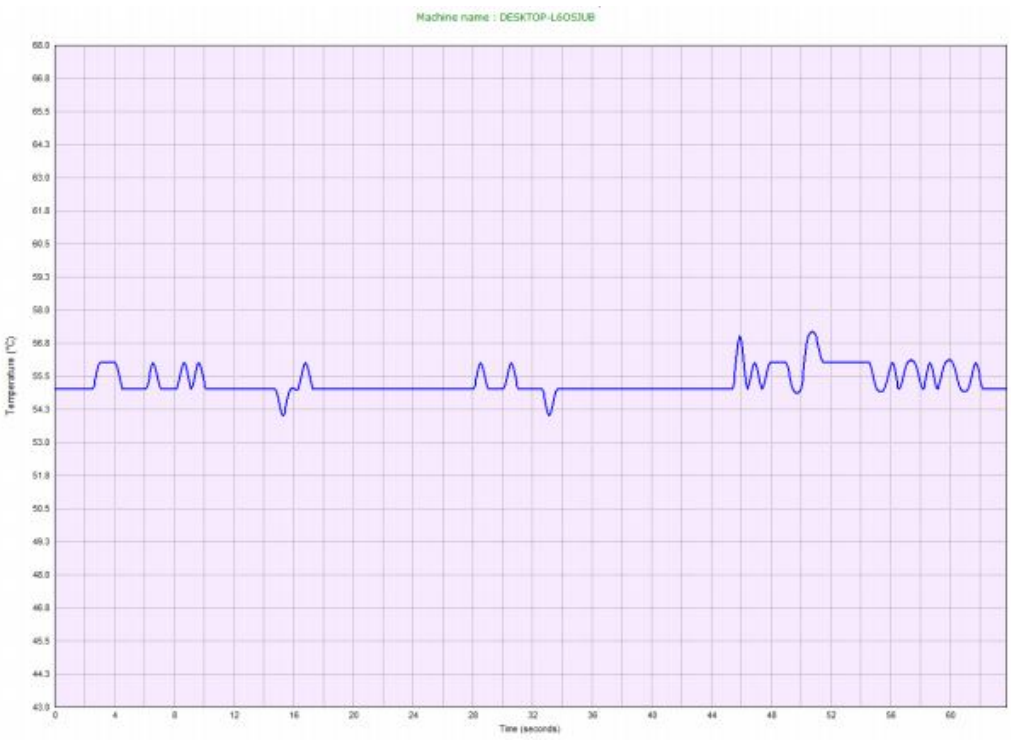

Figure 6.8: Attack 7: Temperature (in deg C) vs Time (in seconds): Core #0

6.3 Phase 3

The table below provides average values for all the parameters P1 to P6 against the defense strategies D1 to D9 scans. Some output graphs are provided as proof of work.
Blue Color indicates that the parameters return lower values compared to that of Phase 1. Black Color indicates that the parameters return higher values compared to that of Phase

Table 2: Attack 1

	D1	D2	D3	D4	D5	D6	D7	D8	D9
P1	38.15	45.5	45.24	45.13	47.35	52.50	51.16	44.72	45.97
P2	2003.0	993.4	587.6	664.7	709.3	883.7	569.3	661.9	1335.0
P3	11.6	14.4	15.0	14.7	10.5	12.3	17.1	13.3	11.7
P4	56.52	44.45	37.57	48.25	53.23	56.3	50.51	53.17	54.9
P5	0.782	0.45	0.991	0.552	0.47	0.791	0.889	0.811	0.779
P6	693.3	650.9	687.3	709	673.43	663.3	687.3	709	763.5

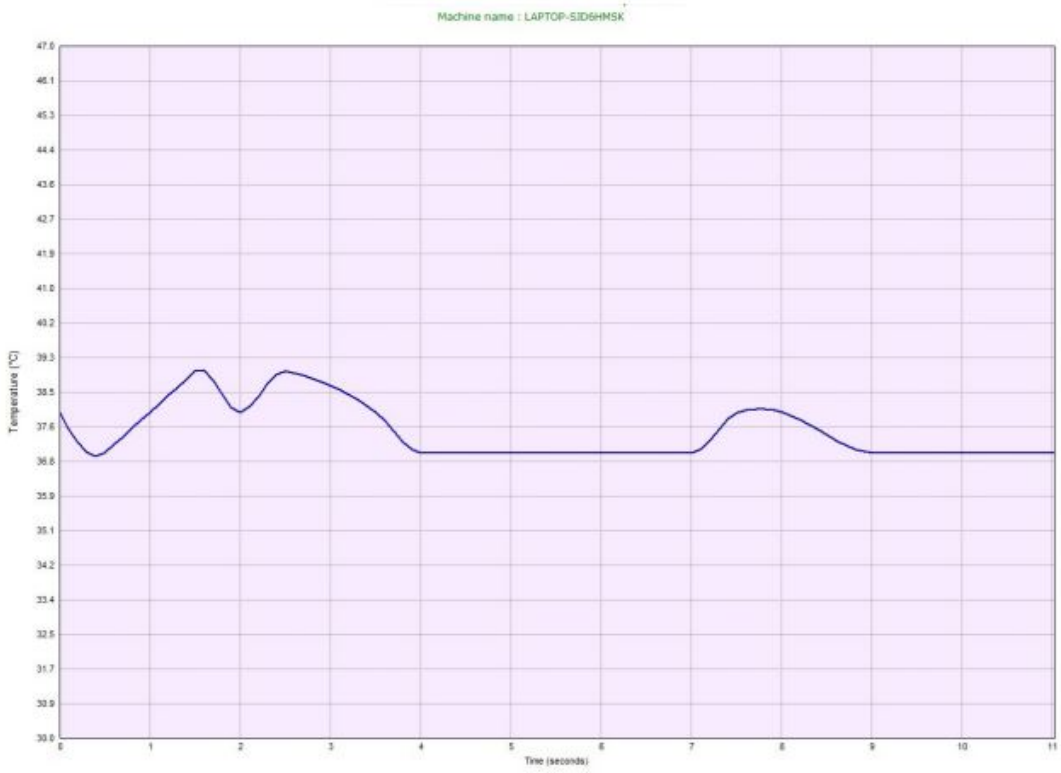

Figure 6.9: Defense 1: Temperature (in deg C) vs Time (in seconds): Core #0

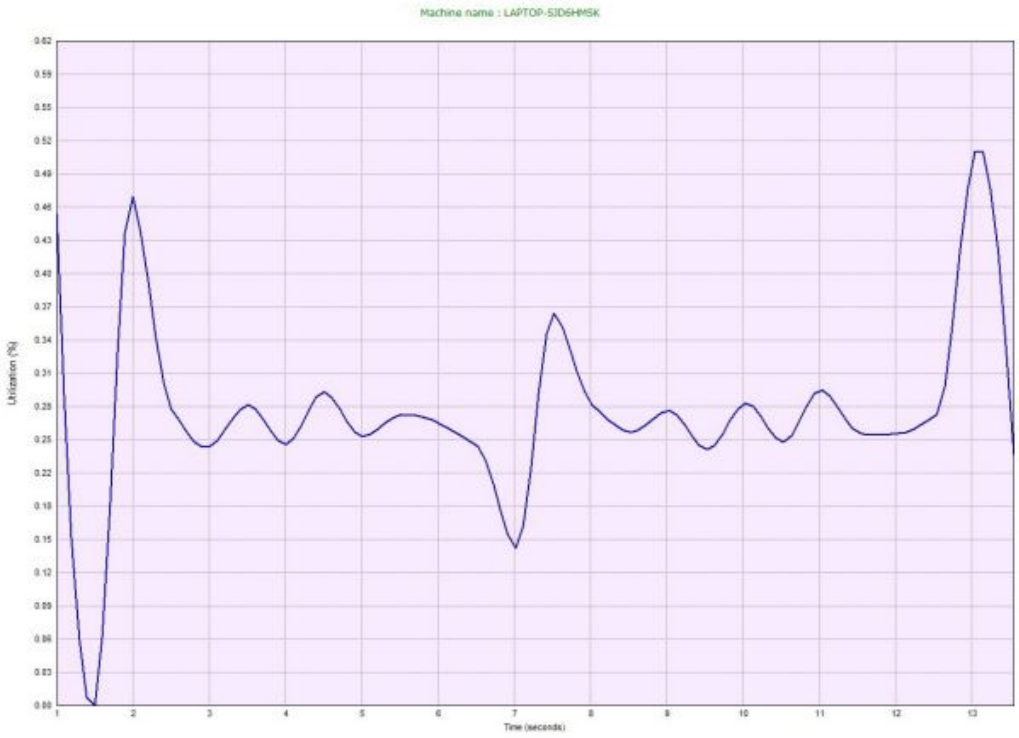

Figure 6.10: Defense 2: HD Graphics GPU Utilization (in %) vs Time (in seconds):Core #0

Figure 6.11: Defense 3: CPU Utilization (in %) vs Time (in seconds):Core #0

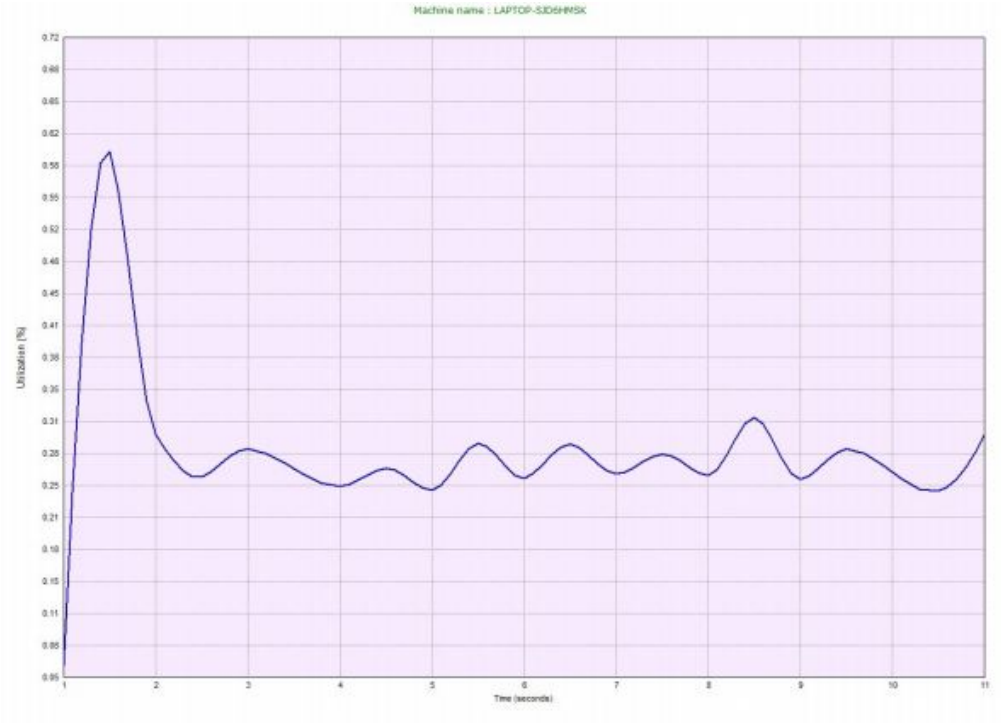

Figure 6.12: Defense 4: CPU Utilization (in %) vs Time (in seconds):Core #0

Table 3: Attack 2

•	D1	D2	D3	D4	D5	D6	D7	D8	D9
P1	49.30	39.90	42.75	52.25	57.50	51.05	52.10	54.75	51.21
P2	1911.0	1323.0	664.5	773.5	783.6	809.2	626.5	1445.3	1995.0
P3	12.7	15.6	11.7	9.3	5.9	12.3	15.6	13.3	12.9
P4	56.11	52.23	54.67	49.67	51.33	54.3	48.80	53.45	45.25
P5	0.78	0.67	0.93	0.66	0.56	0.78	0.87	0.36	0.77
P6	668.4	621.3	762.5	669.3	673.5	641	632	731.2	696.4

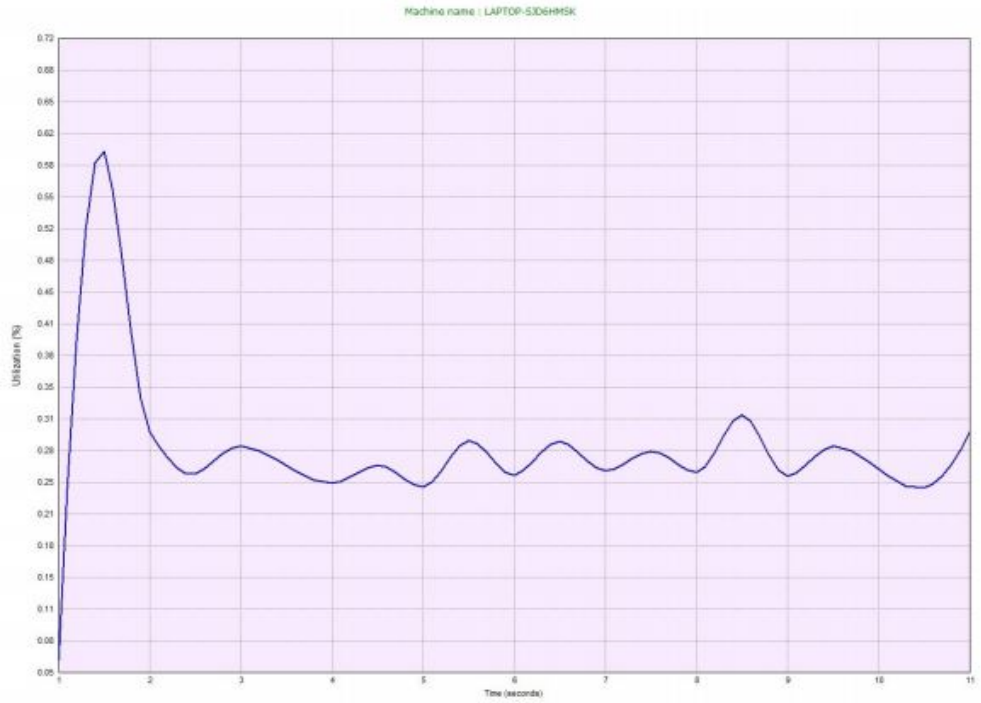

Figure 6.13: Defense 5: HD Graphics GPU Utilization (in %) vs Time (in seconds):Core #0

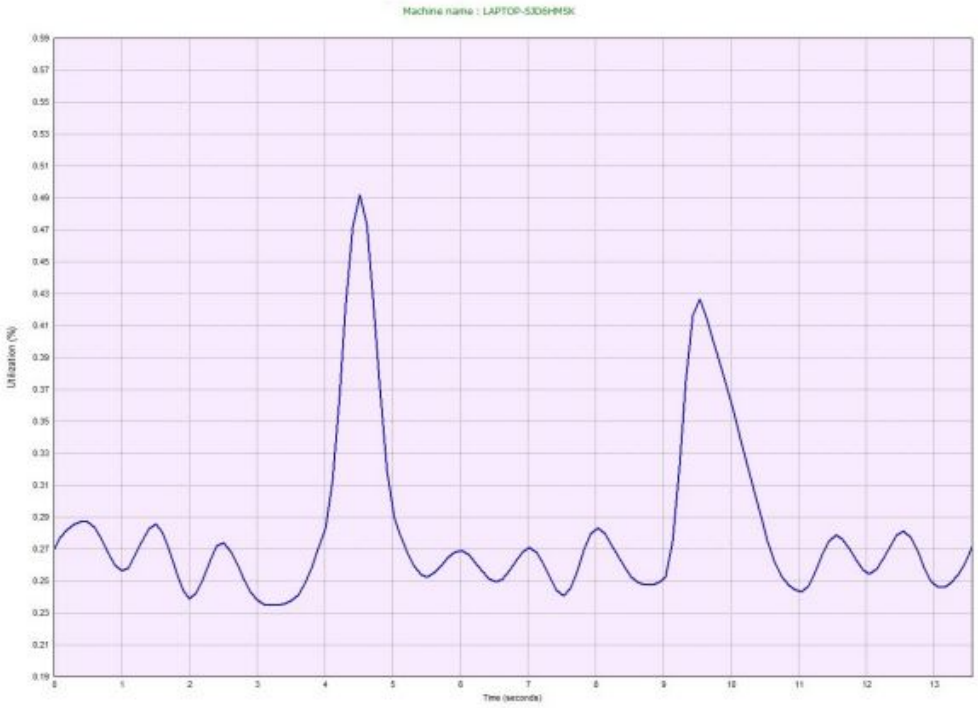

Figure 6.14: Defense 3: HD Graphics GPU Utilization (in %) vs Time (in seconds):Core #0

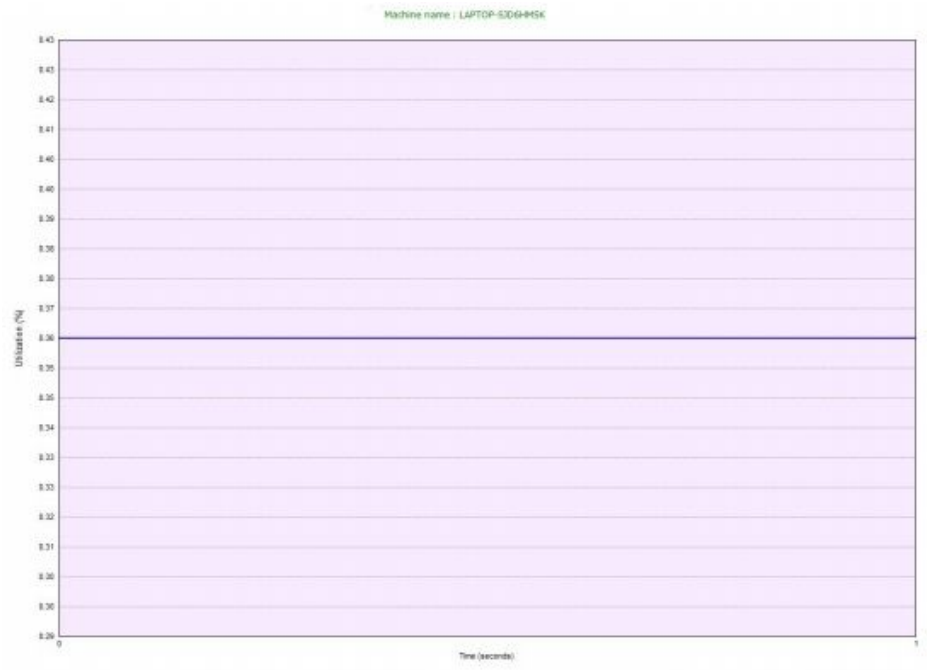

Figure 6.15: Defense 8: HD Graphics GPU Utilization (in %) vs Time (in seconds):Core #0

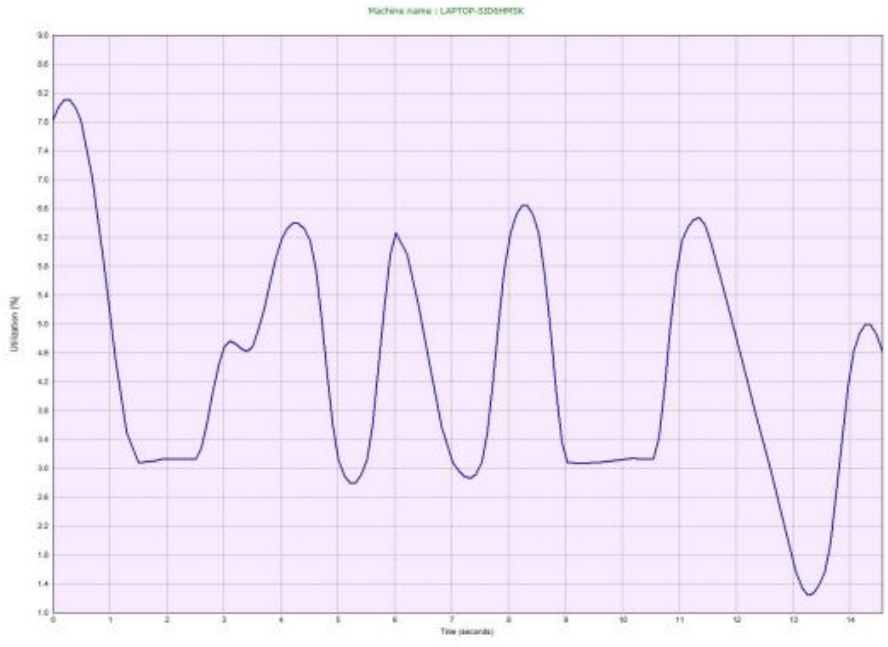

Figure 6.16: Defense 5: CPU Utilization (in %) vs Time (in seconds):Core #0

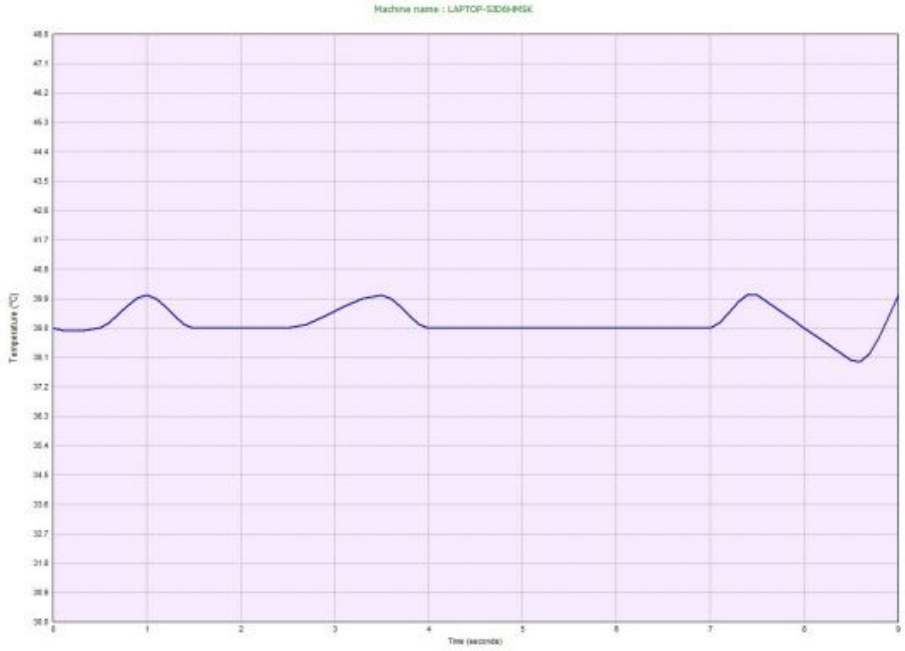

Figure 6.17: Defense 4: Temperature (in deg C) vs Time (in seconds): Core #0

Table 4: Attack 3

•	D1	D2	D3	D4	D5	D6	D7	D8	D9
P1	55.40	42.23	45.15	49.56	51.19	44.90	39.0	53.25	47.25
P2	1325.0	2095.0	662.5	586.5	883.7	1995.0	591.0	709.3	1335.0
P3	15.5	11.7	12.9	9.7	10.5	8.9	13.6	7.9	12.6
P4	55.23	47.67	53.90	49.65	51.25	50.25	48.95	47.66	54.5
P5	0.67	0.34	0.44	0.678	0.699	0.587	0.807	0.906	0.787
P6	621.3	650.3	673.5	663.3	764.5	763.5	693.3	623.8	760.5

Figure 6.18: Defense 9: CPU Clock speed (in %) vs Time (in seconds)

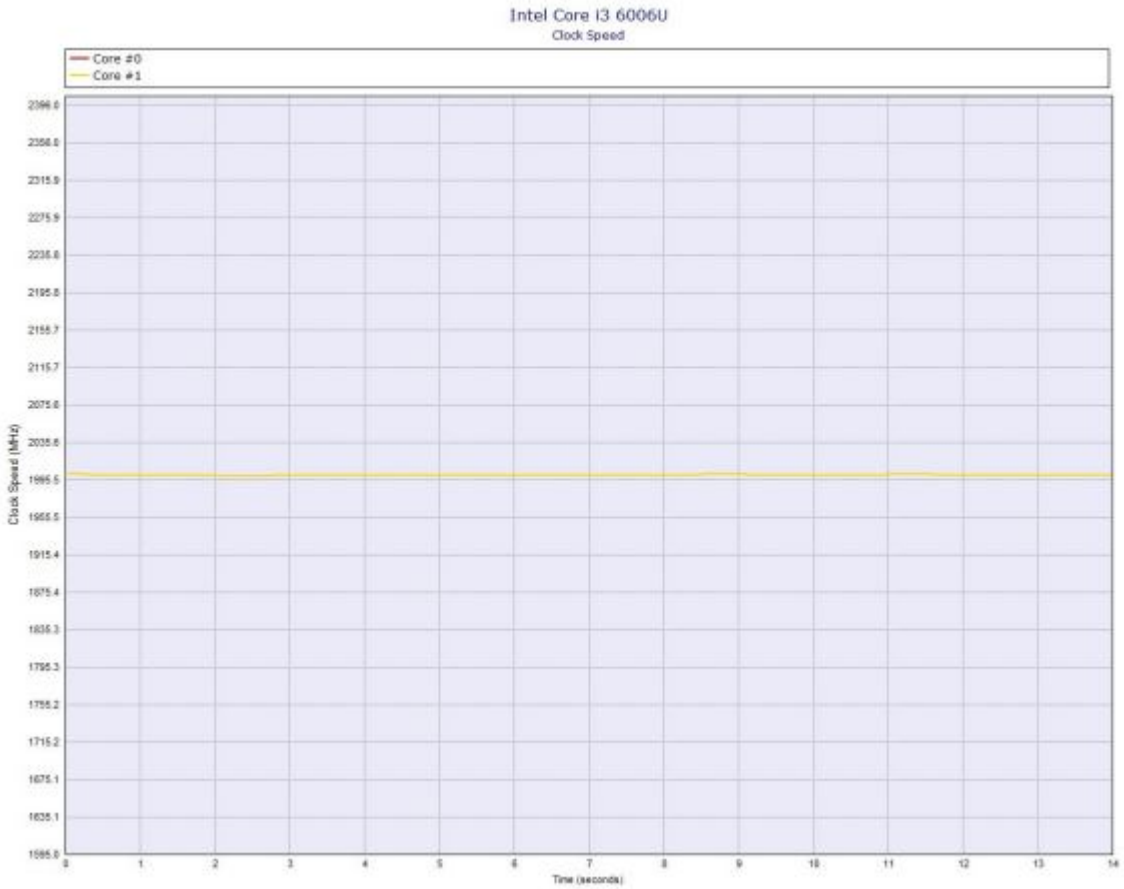

Figure 6.19: Defense 6: HD Graphics Clock speed (in %) vs Time (in seconds)

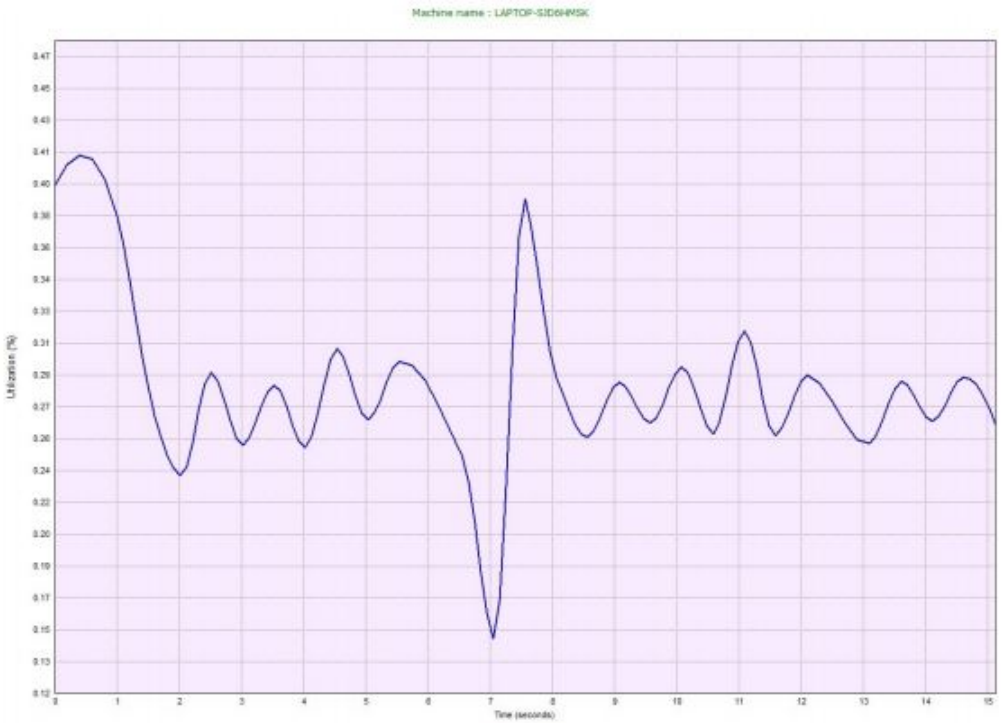

Figure 6.20: Defense 8:Defense 2: HD Graphics Clock speed (in %) vs Time (in seconds)

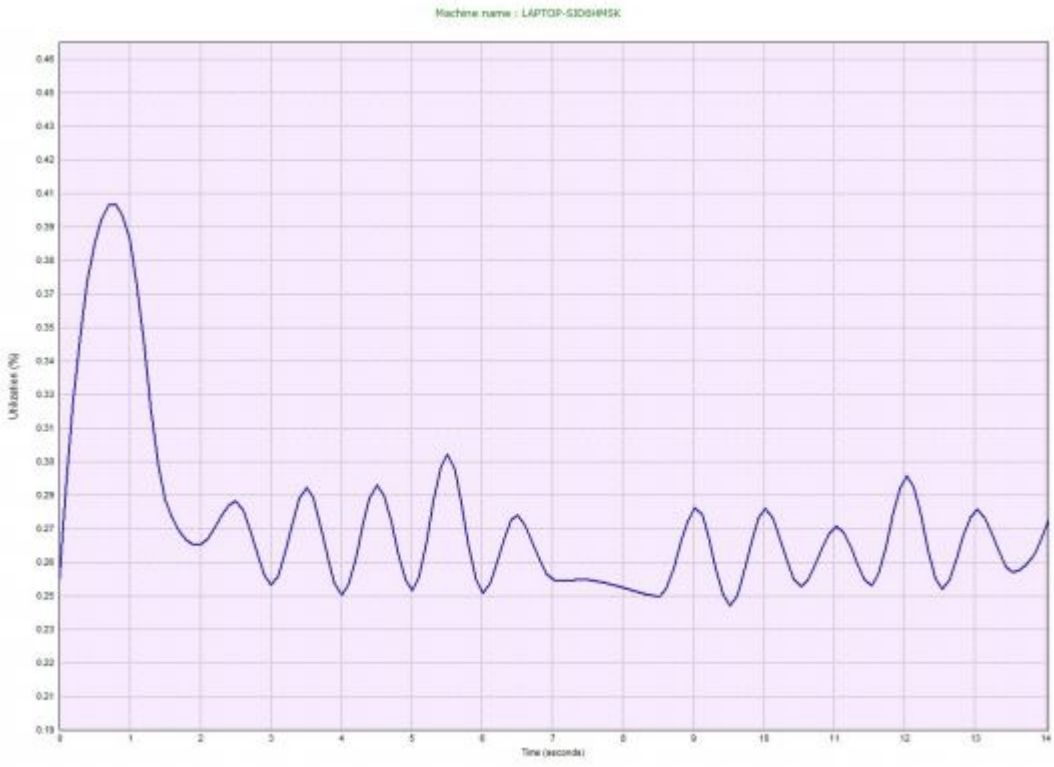

Figure 6.21: Defense 5: CPU Utilization (in %) vs Time (in seconds):Core #0

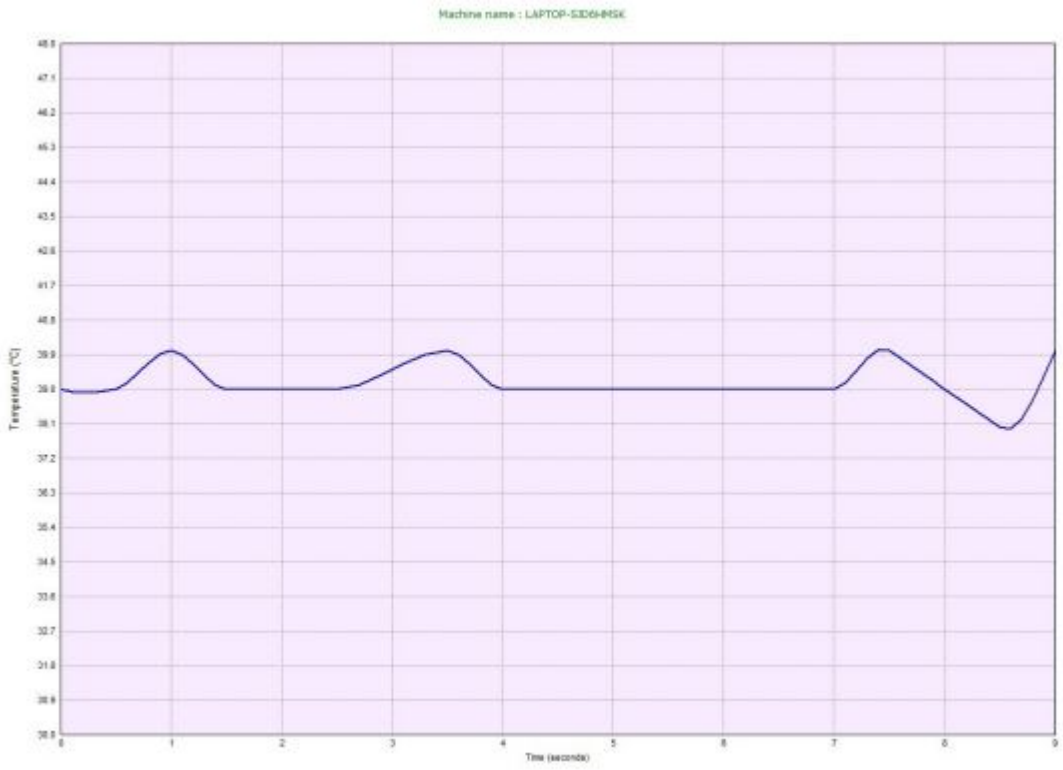

Figure 6.22: Defense 1: HD Graphics GPU Utilization (in %) vs Time (in seconds):Core #0

Table 5: Attack 4

•	D1	D2	D3	D4	D5	D6	D7	D8	D9
P1	49.50	50.90	40.60	50.90	41.45	41.60	54.5	41.2	45.70
P2	662.5	773.5	1075.0	664.7	664.5	809.2	1397.0	783.6	1995.0
P3	14.4	15.5	12.6	11.7	13.3	9.7	9.3	12.9	15.6
P4	54.67	56.10	52.23	54.3	53.23	53.45	50.25	51.25	48.90
P5	0.678	0.782	0.991	0.552	0.67	0.791	0.336	0.93	0.782
P6	693.3	650.9	687.3	709	673.43	663.3	687.3	709	763.5

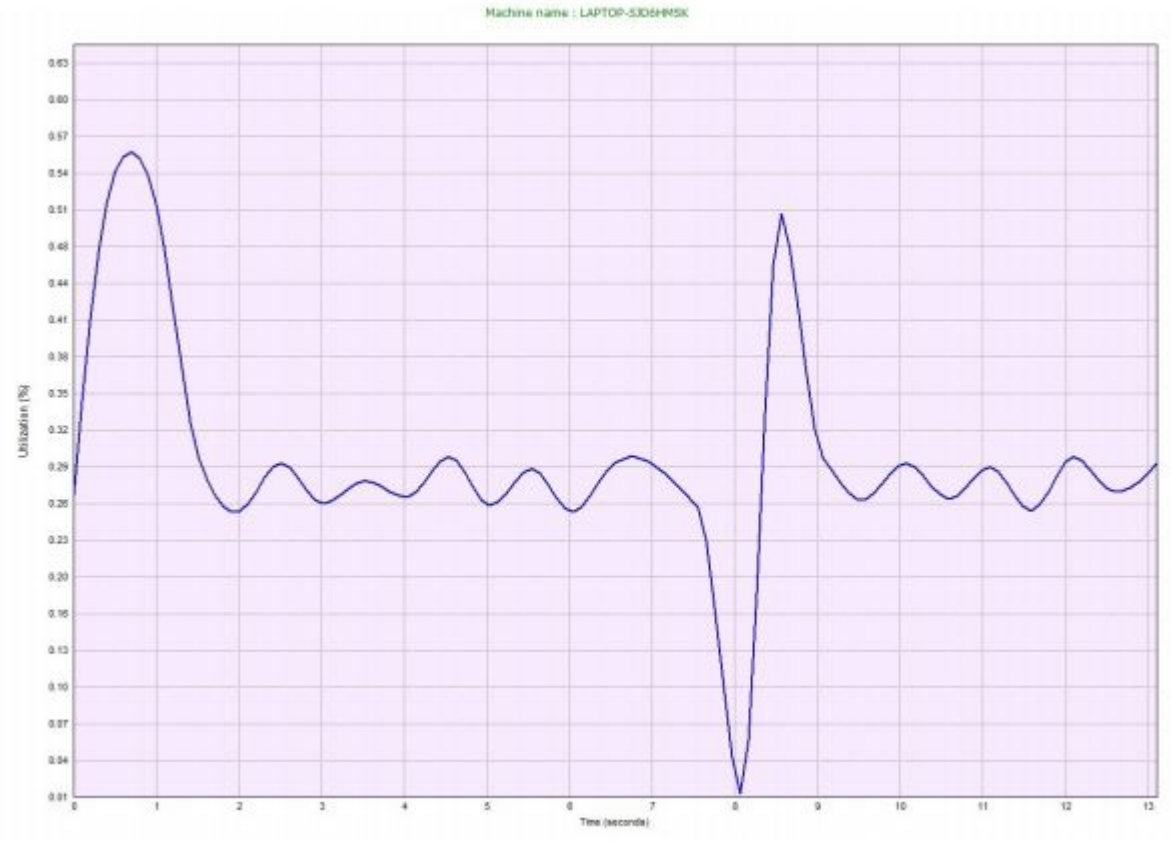

Figure 6.23: Defense 3: HD Graphics GPU Utilization (in %) vs Time (in seconds):Core #0

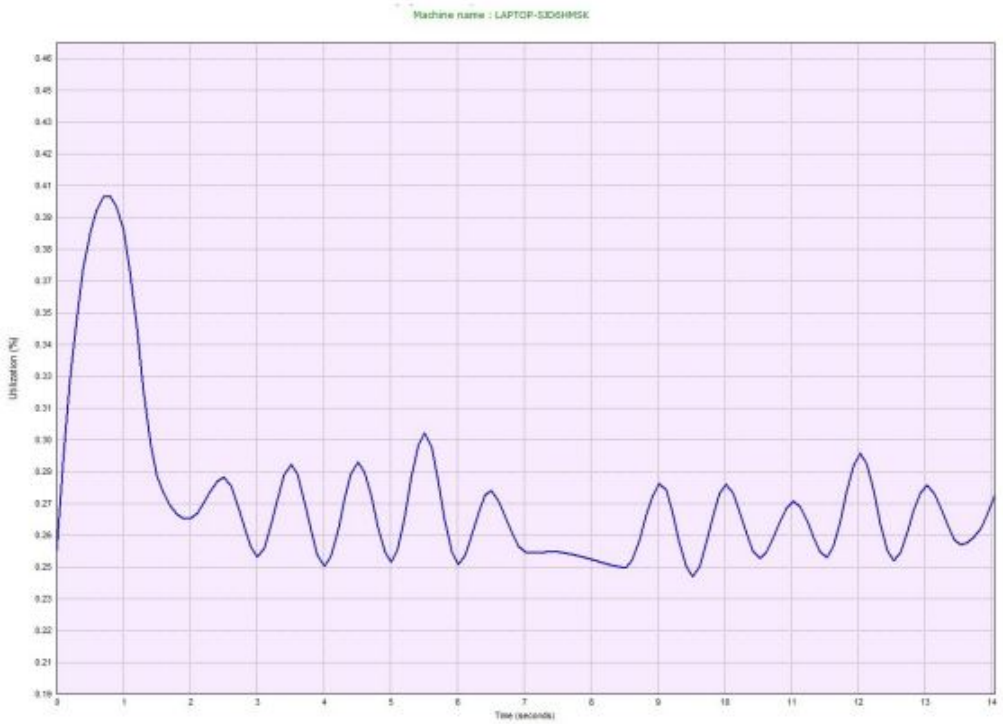

Figure 6.24: Defense 7: HD Graphics GPU Utilization (in %) vs Time (in seconds):Core #0

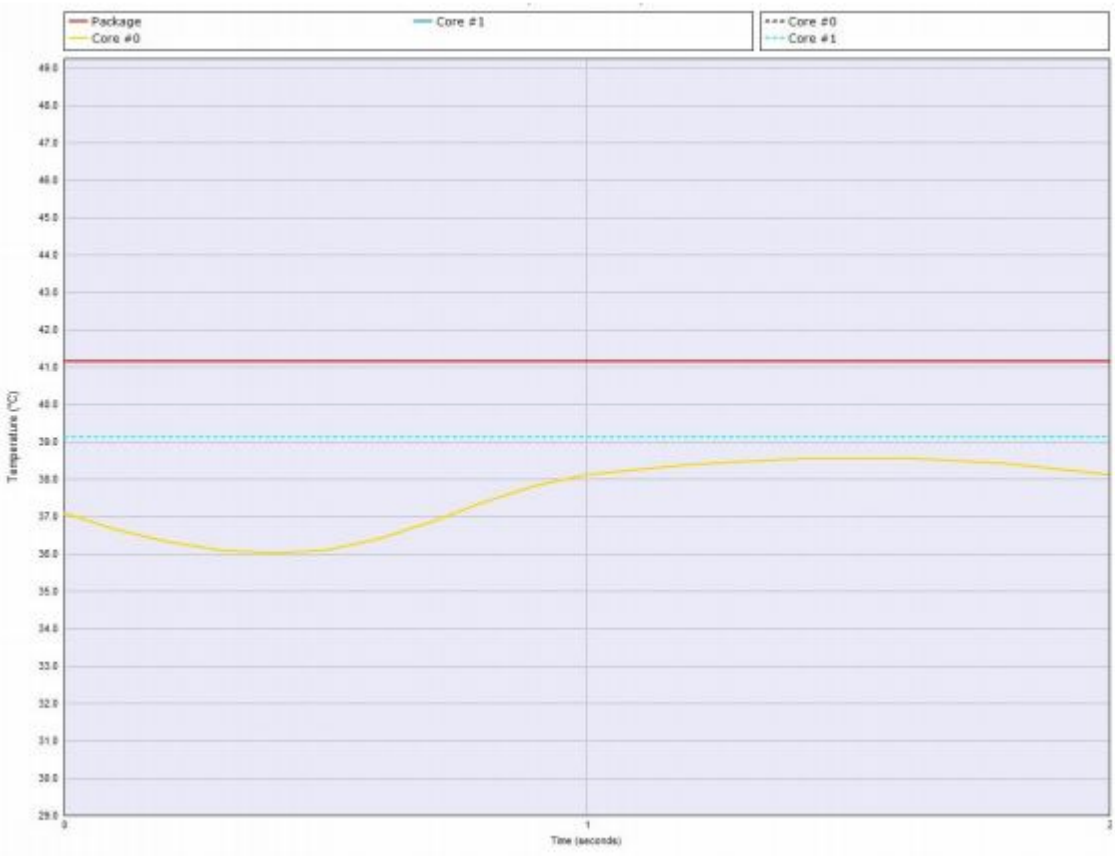

Figure 6.25: Defense 8: Temperature (in deg C) vs Time (in seconds): Core #0, Core #1, Core #2

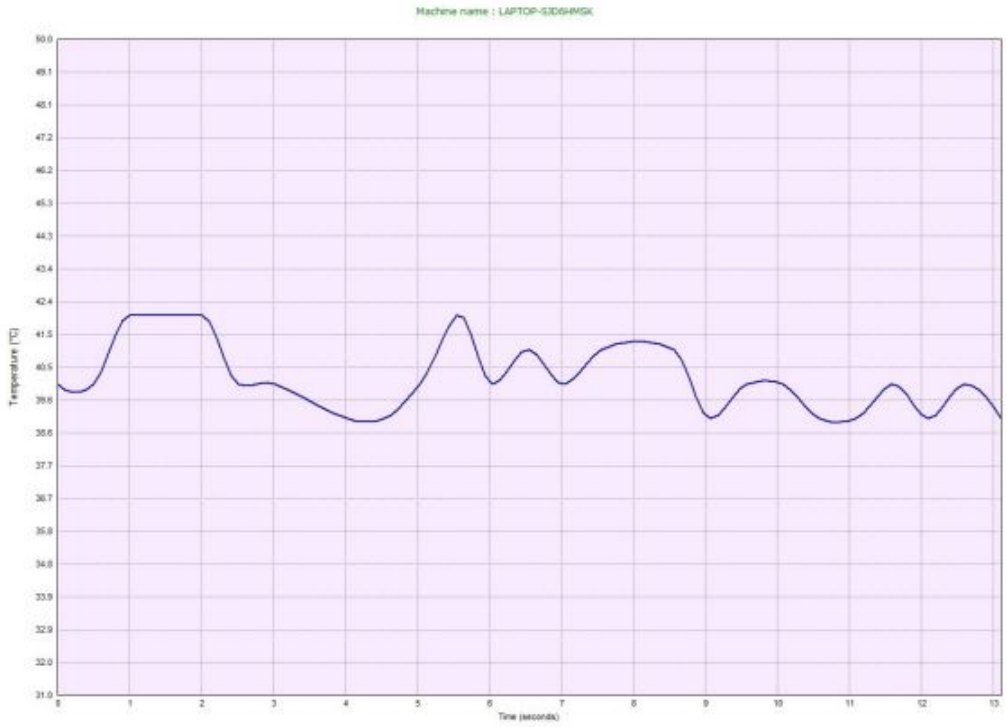

Figure 6.26: Defense 5: Temperature (in deg C) vs Time (in seconds): Core #0

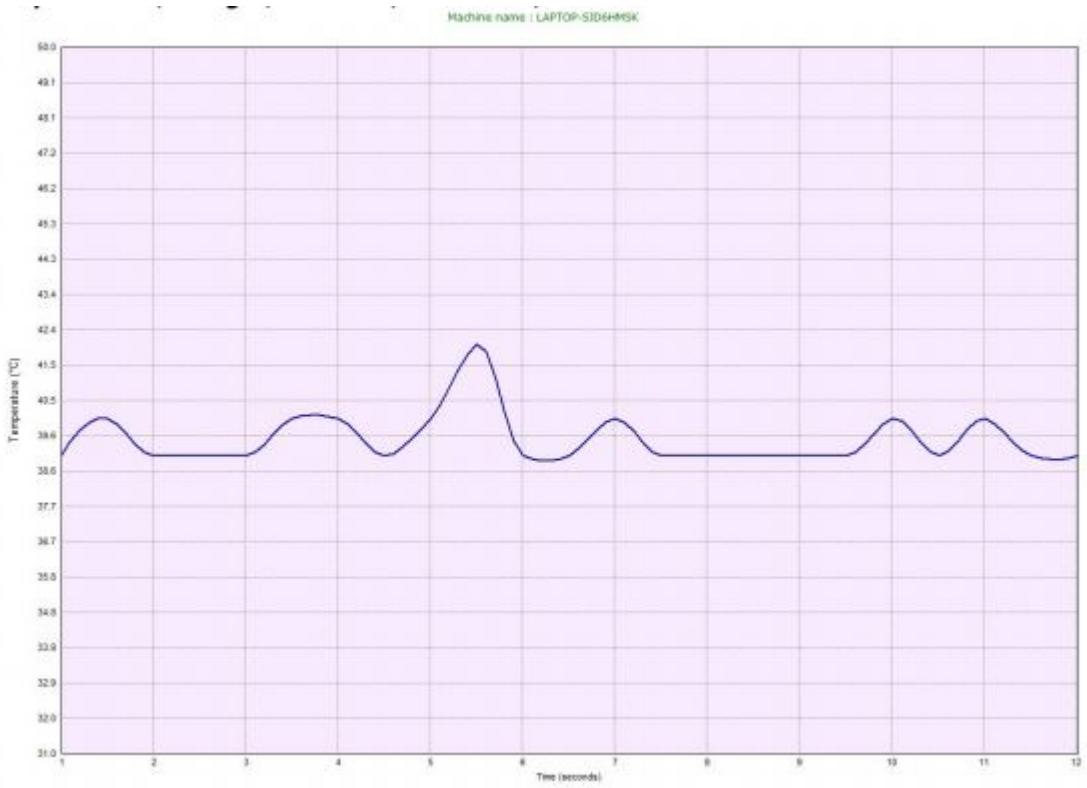

Figure 6.27: Defense 3: Temperature (in deg C) vs Time (in seconds): Core #0

Table 6: Attack 5

•	D1	D2	D3	D4	D5	D6	D7	D8	D9
P1	39.25	50.90	47.35	49.70	56.28	51.05	42.10	40.9	51.26
P2	709.3	993.4	706.5	662.5	709.3	883.7	569.3	661.9	1155.5
P3	7.9	15.3	10.5	14.9	2.5	2.7	5.6	13.3	12.8
P4	47.66	54.45	53.23	54.67	53.23	56.3	50.51	53.17	54.9
P5	0.906	0.782	0.67	0.678	0.67	0.791	0.889	0.811	0.779
P6	623.8	650.9	673.43	693.3	673.43	663.3	687.3	797.8	554.7

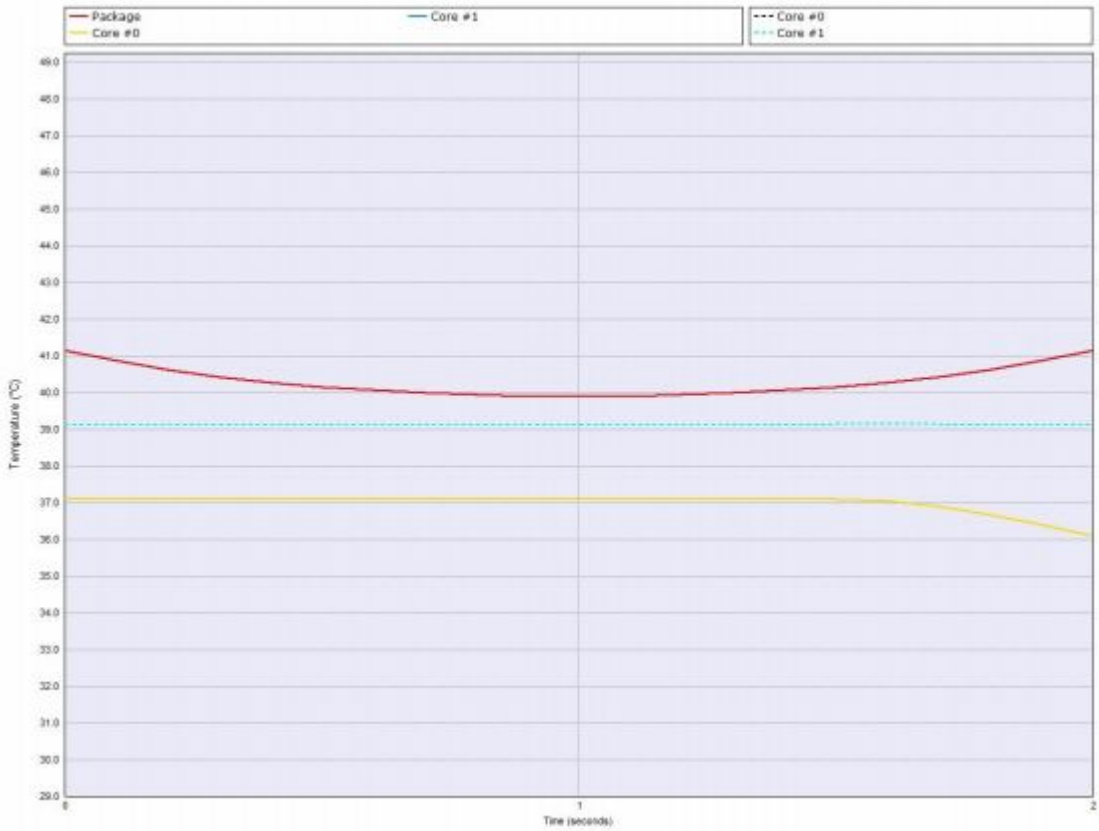

Figure 6.28: Defense 6: Temperature (in deg C) vs Time (in seconds): Core #0, Core #1, Core #2

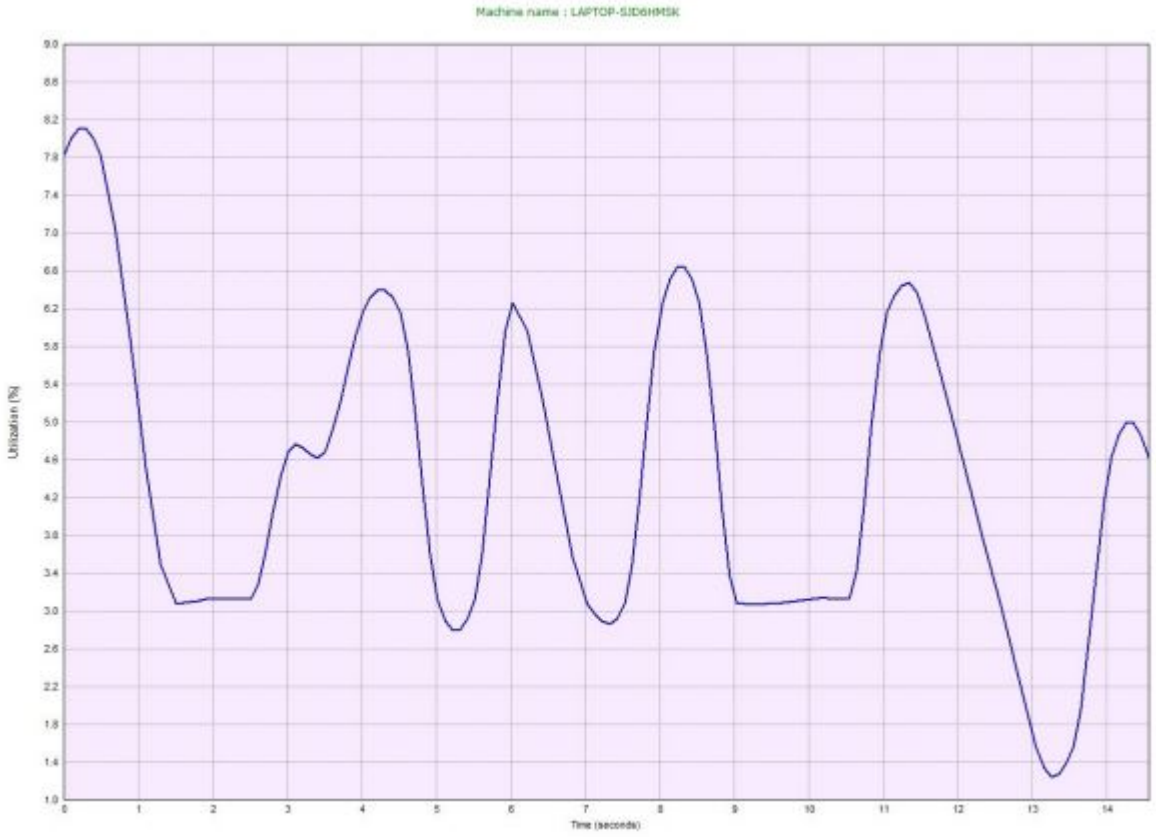

Figure 6.29: Defense 7: CPU Utilization (in %) vs Time (in seconds):Core #0

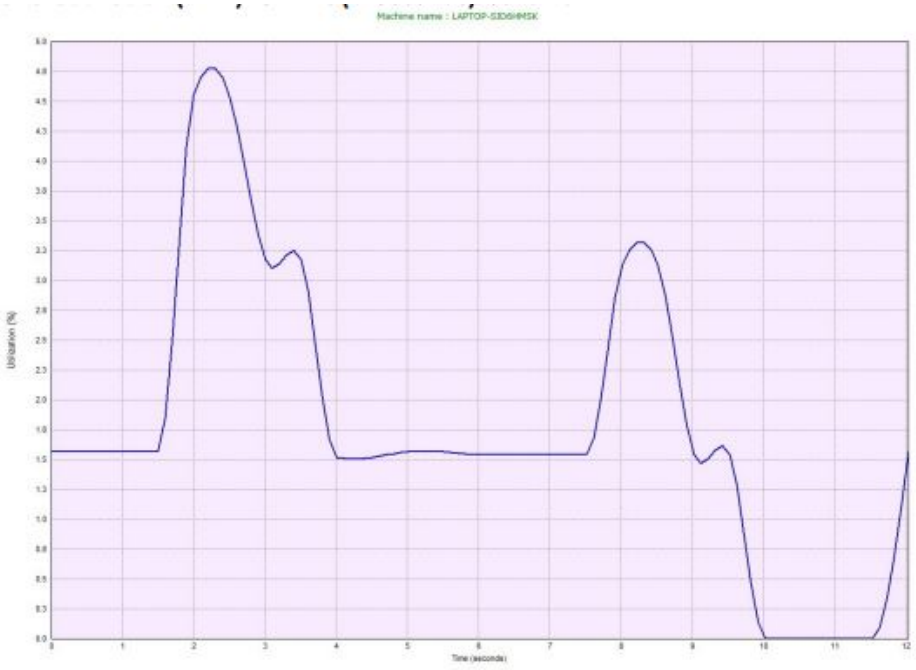

Figure 6.30: Defense 5: CPU Utilization (in %) vs Time (in seconds):Core #0

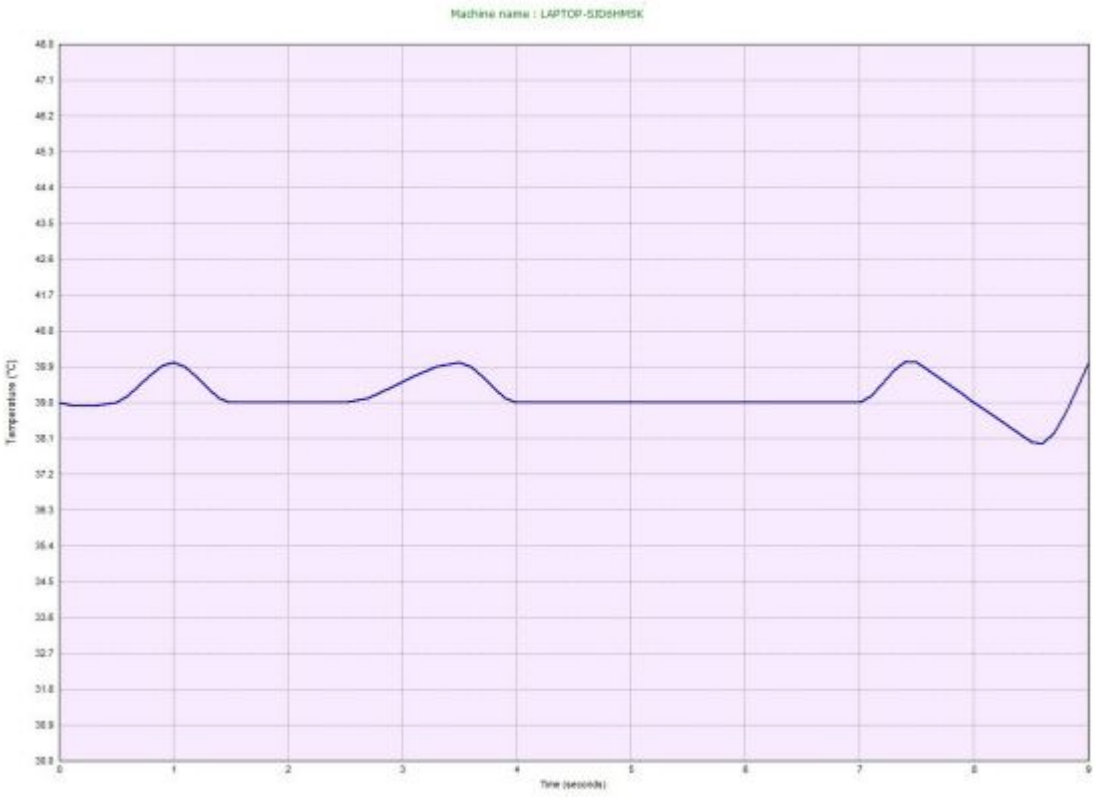

Figure 6.31: Defense 1: Temperature (in deg C) vs Time (in seconds): Core #0

Figure 6.32: Defense 9: CPU Utilization (in %) vs Time (in seconds):Core #0

Table 7: Attack 6

•	D1	D2	D3	D4	D5	D6	D7	D8	D9
P1	49.30	50.90	51.16	42.25	41.50	51.05	50.90	44.75	38.0
P2	685.5	993.4	569.3	664.7	739.3	1995.0	1169.0	626.3	709.3
P3	11.6	6.5	7.3	14.7	10.5	12.3	15.6	13.3	10.5
P4	56.52	54.45	50.51	50.25	53.23	56.3	52.23	53.17	53.23
P5	0.32	0.875	0.889	0.409	0.63	0.761	0.67	0.811	0.67
P6	693.3	632.6	687.3	722.6	671.6	605,6	621.3	709.6	673.4

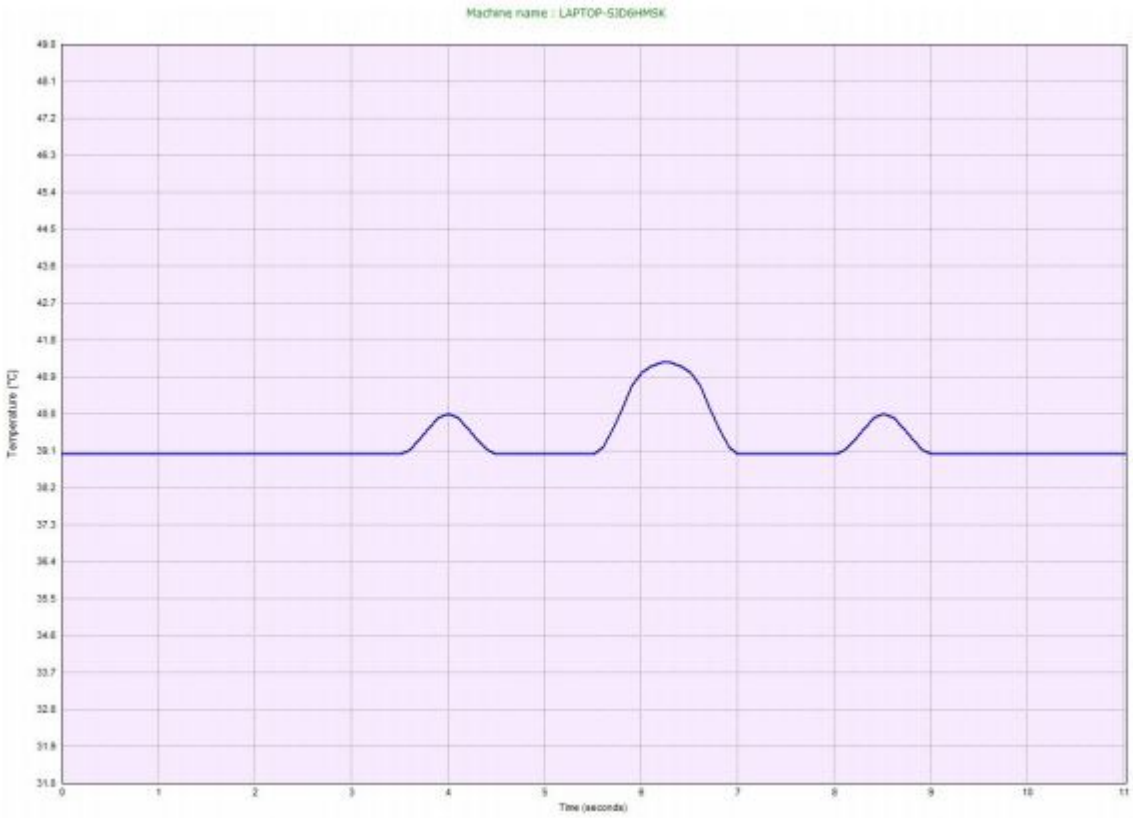

Figure 6.33: Defense 8: Temperature (in deg C) vs Time (in seconds): Core #0

Figure 6.34: Defense 6: HD Graphics Clock speed (in %) vs Time (in seconds)

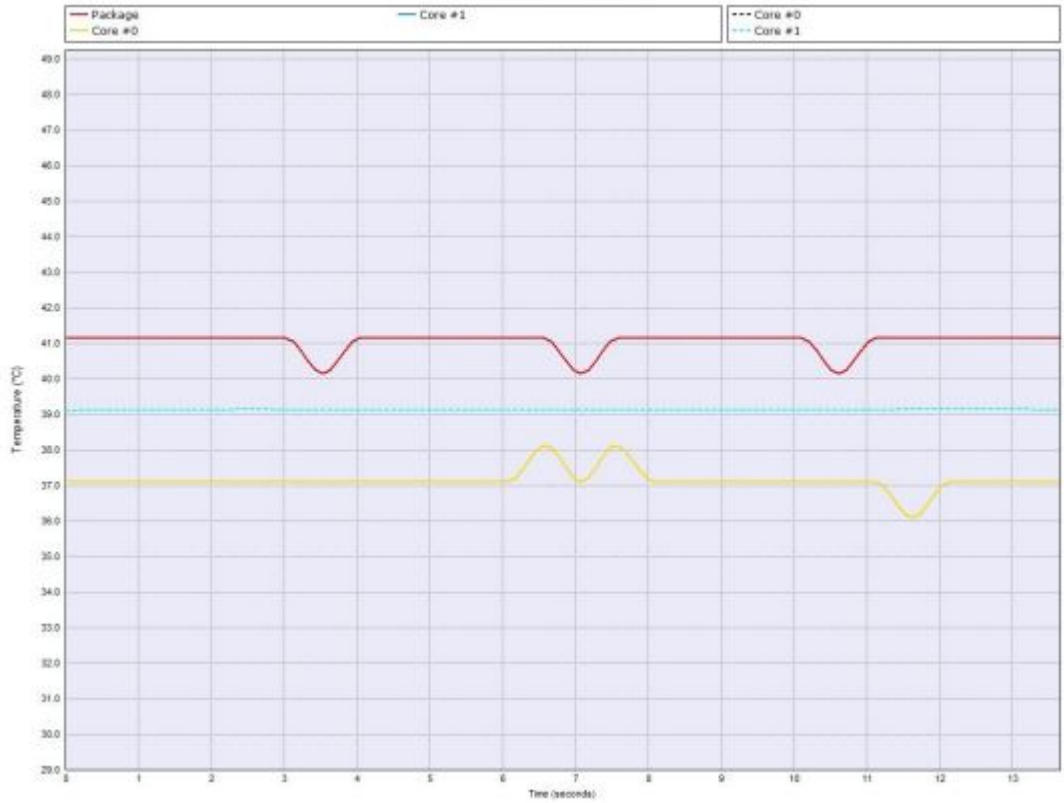

Figure 6.35: Defense 5: Temperature (in deg C) vs Time (in seconds): Core #0

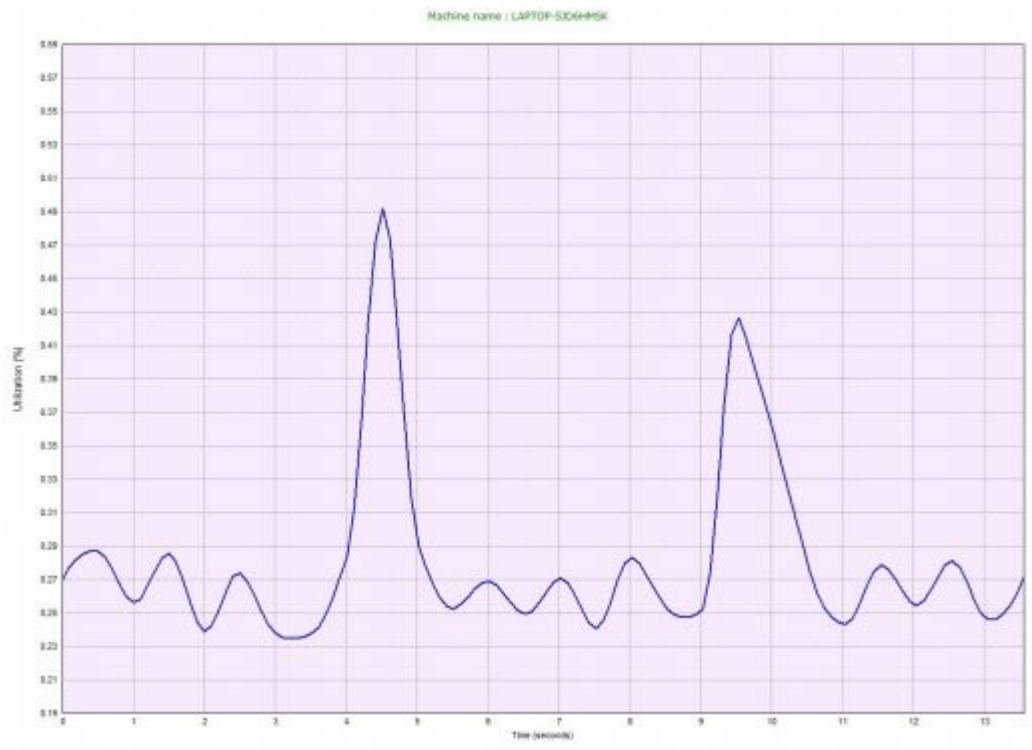

Figure 6.36: Defense 5: HD Graphics GPU Utilization (in %) vs Time (in seconds):Core #0

Figure 6.37: Defense 4: Temperature (in deg C) vs Time (in seconds): Core #0

Table 8: Attack 7

•	D1	D2	D3	D4	D5	D6	D7	D8	D9
P1	51.05	51.20	50.90	52.25	47.50	41.05	42.10	49.50	39.80
P2	809.2	1083.7	664.7	664.7	772.5	862.7	865.5	662.5	557.6
P3	12.3	10.6	4.4	14.7	2.5	1.7	2.7	6.4	9.7
P4	54.3	54.45	54.3	50.25	53.23	56.3	50.51	54.67	54.9
P5	0.76	0.782	0.37	0.552	0.67	0.791	0.889	0.678	0.779
P6	641	650.9	737.9	879.7	673.43	663.3	662.3	693.3	753.5

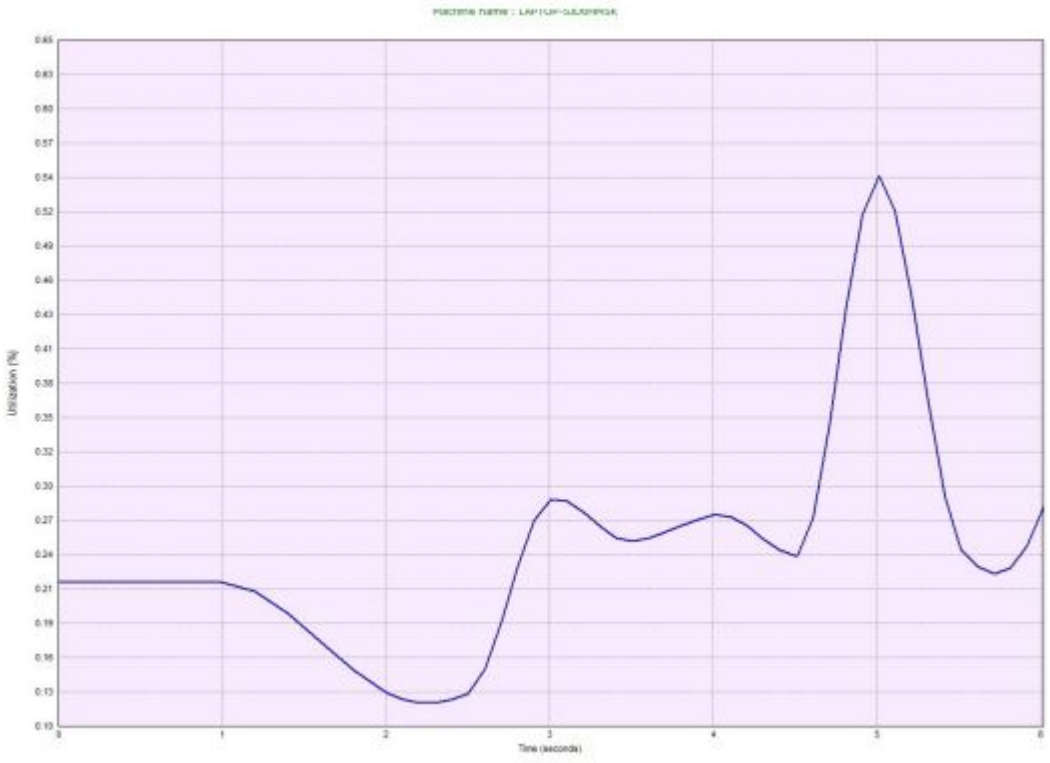

Figure 6.38: Defense 1: HD Graphics GPU Utilization (in %) vs Time (in seconds):Core #0

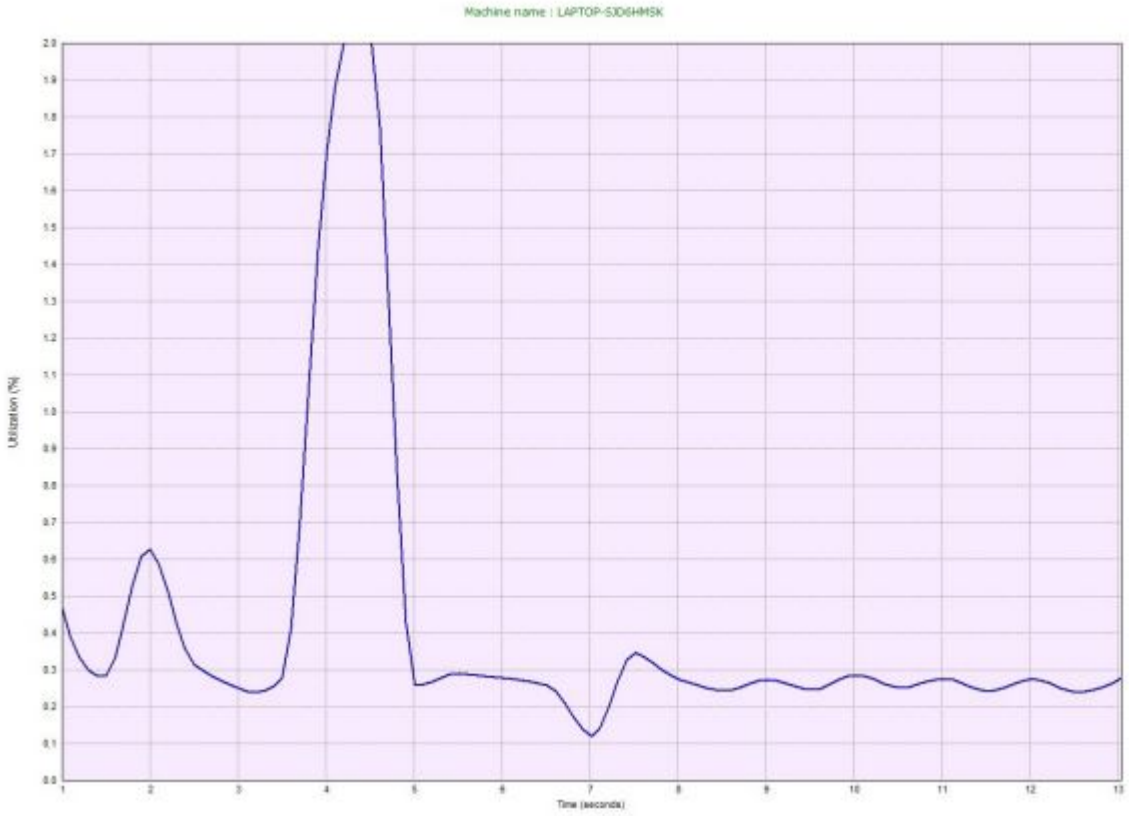

Figure 6.39: Defense 5: CPU Utilization (in %) vs Time (in seconds):Core #0

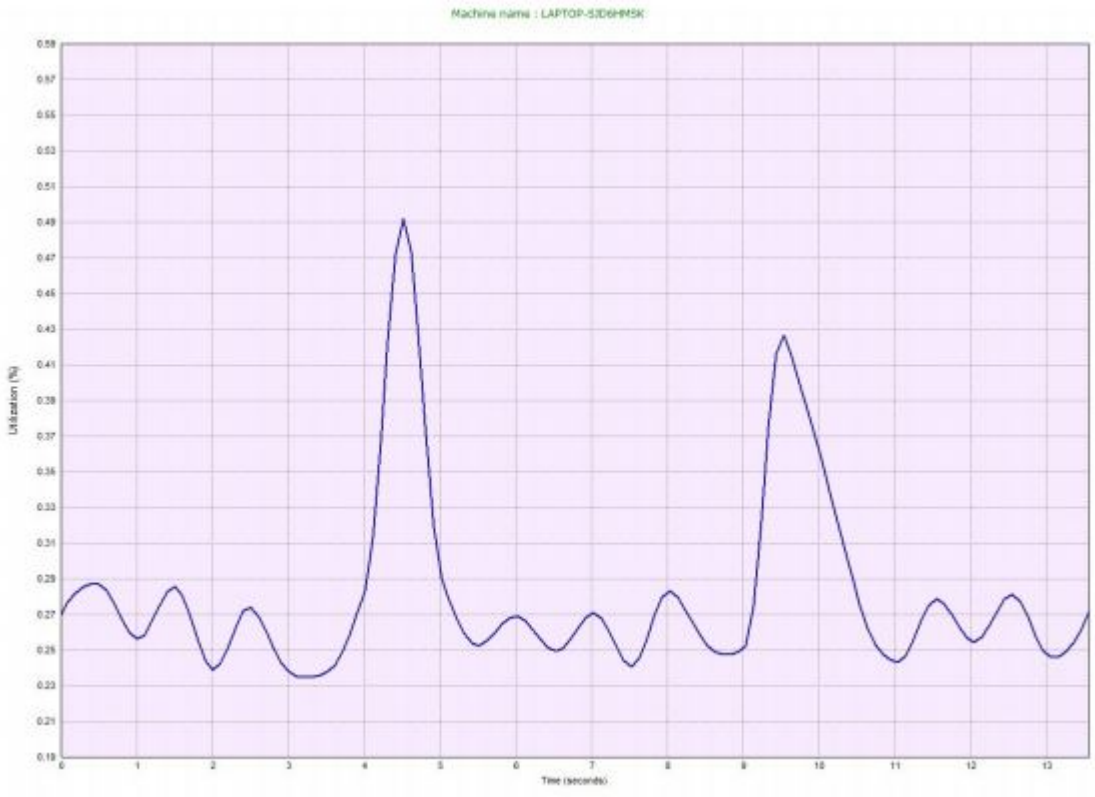

Figure 6.40: Defense 8: HD Graphics GPU Utilization (in %) vs Time (in seconds):Core #0

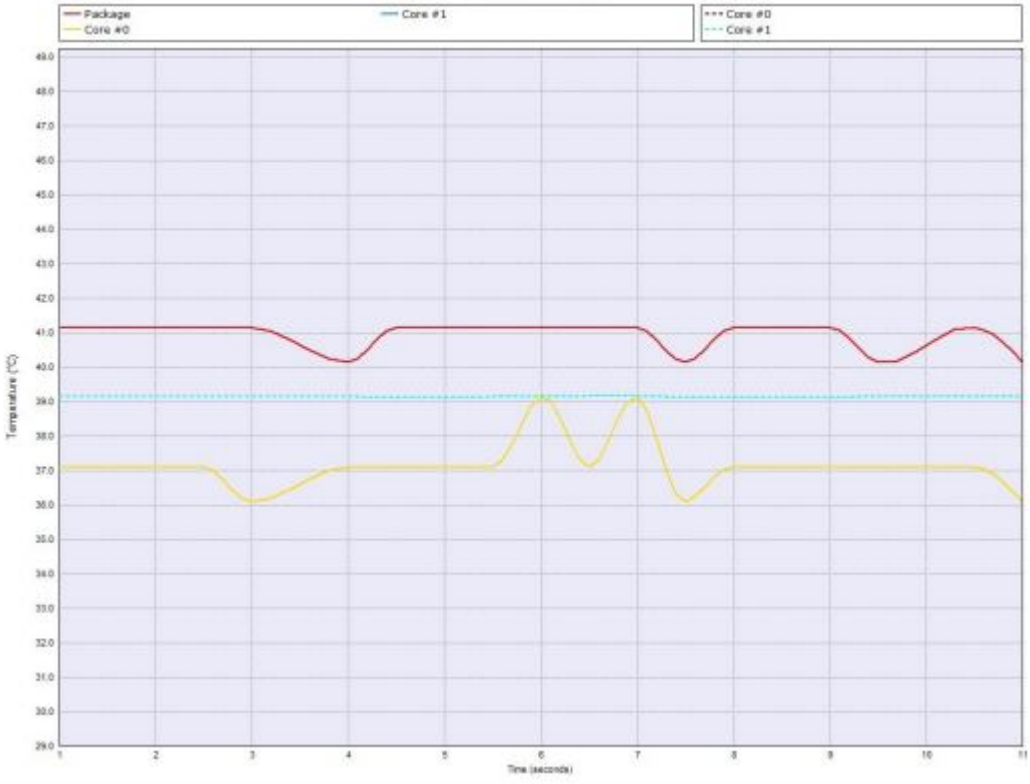

Figure 6.41: Defense 6: Temperature (in deg C) vs Time (in seconds): Core #0

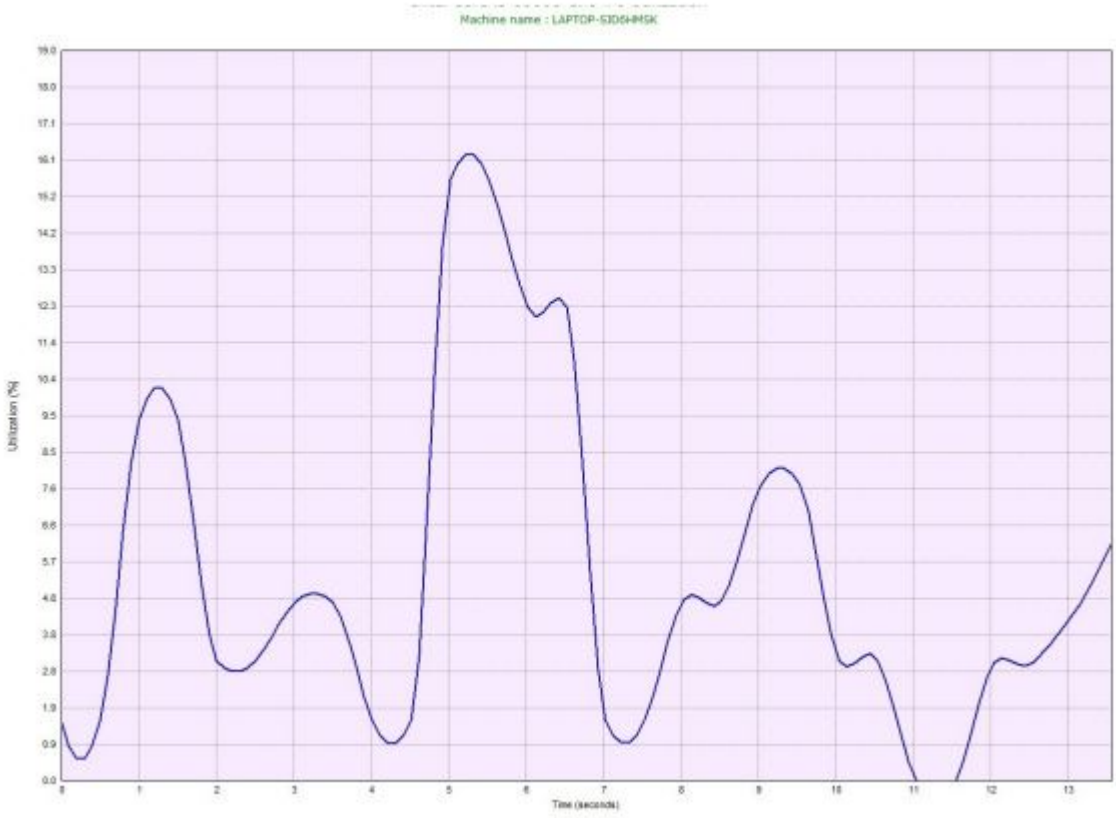

Figure 6.42: Defense 2: CPU Utilization (in %) vs Time (in seconds):Core #0

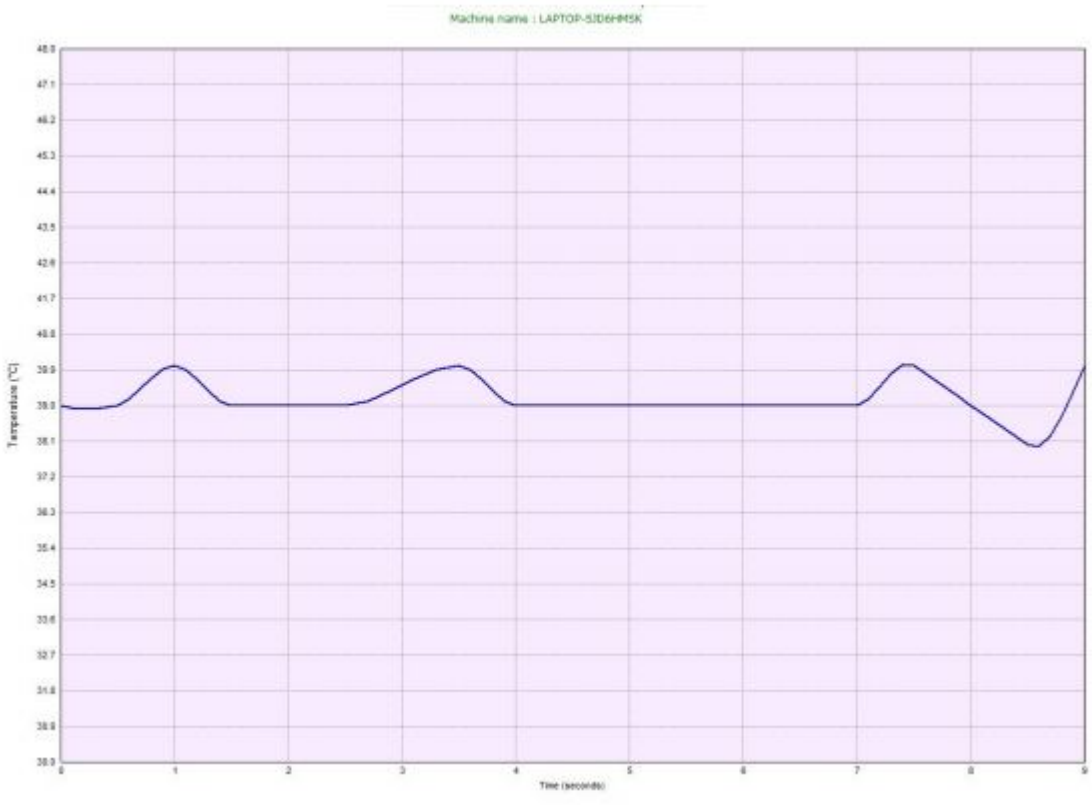

Figure 6.43: Defense 9: Temperature (in deg C) vs Time (in seconds): Core #0

Chapter 7

Results and discussion

7.1 Attack Report

The attacks carried out by the kali system were basic. Every attack performed on the defence system enabled different levels of client system compromises. The list of the attacks with the session details goes as below.

- Zenmap Attack:
  The attack simply retrieved the OS details of the host system. This attack had different categories. Intense scanning was unfortunately reported at the defence system. Rest of the scans had no issue retrieving the details.
- Spear Phishing:
  This attack generates a payload, a mass mailer attack, which was attached with a reverse TCP payload. The payload was not very successful at causing the reverse connection every time, but did achieve to cause a load on the host system.
- Backdoor (Simple):
  This attack generates a payload, a reverse TCP payload, which successfully causes a reverse connection back to the attacker at all times. One observations worth mentioning during this attack is, if the payload gets deleted after it is once executed, the attacker still has access to the host system.
- Backdoor (Meterpreter):
  This attack generates a payload, a reverse TCP payload, which successfully causes a reverse connection back to the attacker at all times. An added functionality is that, one can hide this process into another process, and execute it upon booting, with the other process.

- EvilGrade:
  EvilGrade is a type of Man-in-the-Middle attack, with extra functionalities. The intention was a simple OS breach, which was achieved.
- Infectious Media Generator:
  This attack produces a infectious media file (of the formats .mp4, .jpg, .bmp, .mp3, etc) which is basically embedded with reverse TCP payload which completes the communication back to the attacker successfully. The media works just fine irrespective of the embedded virus.
- Custom Virus:
  A custom virus which is a folder bomb, was built and executed. It was successful in slowing down the system.

7.2 Chain of Custody

EVIDENCE CHAIN OF CUSTODY TRACKING FORM

Case Number: 1
Offense: _____
Submitting Officer: (Name/ID#) _____
Victim: 192.168.0.3 (Lenovo B40-30, Windows 10 pro) Owner: Rahul Hulli
Suspect: 192.168.14.43 (Lenovo Ideapad 320) Owner: Rahul Hulli
Date/Time Seized: ____August,2018_____
Location of Seizure: _____GEC, Farmagudi_____

Description of Evidence			
Sr#	**Attack**	**Description**	Type Of Compromise
1	Zenmap Attack	The attack simply retrived the OS details of the host system. This attack had different categories. Intense scanning was unfortunately reported at the defense system. Rest of the scans had no issue retrieving the details.	OS Details
2	Spear Phishing	This attack generates a payload, a mass mailer attack, which was attached with a reverse TCP payload. The payload was not very successful at causing the reverse connection everytime, but did acheive to cause a load on the host system.	Access to cmd
3[i]	Backdoor (Simple)	This attack generates a payload, a reverse TCP payload, which successfully causes a reverse connection back to the attacker at all times. One observations worth mentioning during this attack is, if the payload gets deleted after it is once executed, the attacker still has access to the host system.	Access to cmd
4	Backdoor (Meterpreter)	This attack generates a payload, a reverse TCP payload, which successfully causes a reverse connection back to the attacker at all times. An added functionality is that, one can hide this process into another process, and execute it upon booting, with the other process.	Access to cmd
5	EvilGrade	EvilGrade is a type of Man-in-the-Middle attack, with extra functionalities. The intention was a simple OS breach.	OS Breach
6	Infectious Media Generator	This attack produces a infectious media file (of the formats .mp4, .jpg, .bmp, .mp3, etc) which is basically embedded with reverse TCP payload which completes the communication back to the attacker successfully. The media works just fine irrespective of the embedded virus.	Access to cmd
7	Custom Virus	A custom virus which is a folder bomb, was built and executed. It was successful in slowing down the system.	None

7.3 Defence Report

Different defensive strategies were used for the 7 attacks. Every defence strategy caused a compromise reduction against the attacks. The list is as follows.

- Host-based Firewall: Zone Alarm
  - Attack 1 (Zenmap):
    In ZoneAlarm, program access is controlled by way of ”zones”, into which all network connections are divided. The ”trusted zone” generally includes the user’s local area network and can share resources such as files and printers. The while the ”Internet zone” includes everything not in the trusted zone. HBF restricts this attack and notifies the breach, since this software has control over open ports. If the software has been activated earlier, the attack is not possible to be carried out.
  - Attack 2 (Spear Phishing):
    This attack generates a payload, a mass mailer attack, which was attached with a reverse TCP payload. The reverse TCP payload did not work when the HBF was activated.
  - Attack 3 (Backdoor (Simple)):
    This attack generates a payload, a reverse TCP payload, which successfully causes a reverse connection back to the attacker at all times. HBF could restrict this attack, only after a short amount of time. The breach by the attacker was successful, but didn’t go long.
  - Attack 4 (Backdoor (Meterpreter)):
    This attack generates a payload, a reverse TCP payload, which successfully causes a reverse connection back to the attacker at all times. HBF could restrict this attack, only after a short amount of time, but alongside caused troubles to the process with which the payload merged. The breach by the attacker was successful, the attacker still had access for a long time even after HBF was active.
  - Attack 5 (EvilGrade):
    EvilGrade is a type of Man-in-the-Middle attack, with extra functionalities. This attack was totally put down by the HBF system.
  - Attack 6 (Infectious Media Generator):
    This attack produces a infectious media file (of the formats .mp4, .jpg, .bmp, .mp3, etc) which is basically embedded with reverse TCP payload which completes the communication back to the attacker success-

fully. HBF system detected the payload embedded and quarantined the media file.
  - Attack 7 (Custom Virus):
    A custom virus which is a folder bomb, was built and executed. It was successful in slowing down the system. HBF had no effect on this virus, the virus continued to function without harm.

- Web-based Firewall: Comodo Security
  - Attack 1 (Zenmap):
    The attack simply retrieved the OS details of the host system. The defence mechanism totally put down the attack.
  - Attack 2 (Spear Phishing):
    This attack generates a payload, a mass mailer attack, which was attached with a reverse TCP payload. WBF was successful in stopping this attack.
  - Attack 3 (Backdoor (Simple)):
    This attack generates a payload, a reverse TCP payload, which successfully causes a reverse connection back to the attacker at all times. WBF could totally restrict this attack.
  - Attack 4 (Backdoor (Meterpreter)):
    This attack generates a payload, a reverse TCP payload, which successfully causes a reverse connection back to the attacker at all times. WBF detected the attack,but caused harm to the process with which it had merged.
  - Attack 5 (EvilGrade):
    EvilGrade is a type of Man-in-the-Middle attack, with extra functionalities. This attack was totally put down by the WBF system.
  - Attack 6 (Infectious Media Generator):
    This attack produces a infectious media file (of the formats .mp4, .jpg, .bmp, .mp3, etc) which is basically embedded with reverse TCP payload which completes the communication back to the attacker successfully. WBF restricted the reverse communication.
  - Attack 7 (Custom Virus):
    A custom virus which is a folder bomb, was built and executed. It was successful in slowing down the system. WBF had no effect on this virus, the virus continued to function without harm.

- Intrusion Prevention System: Comodo Security

  - Attack 1 (Zenmap):
    The attack simply retrieved the OS details of the host system. The IPS system prevented the breach and notified to the user, hence the attack was totally beaten.
  - Attack 2 (Spear Phishing):
    This attack generates a payload, a mass mailer attack, which was attached with a reverse TCP payload. The IPS system prevented the payload to function and was successfully able to remove it.
  - Attack 3 (Backdoor (Simple)):
    This attack generates a payload, a reverse TCP payload, which successfully causes a reverse connection back to the attacker at all times. The IPS system prevented the backdoor payload function and removed it successfully.
  - Attack 4 (Backdoor (Meterpreter)):
    This attack generates a payload, a reverse TCP payload, which successfully causes a reverse connection back to the attacker at all times. The IPS system prevented the attack and didn't cause a substantial harm to the process to which the payload got merged with.
  - Attack 5 (EvilGrade):
    EvilGrade is a type of Man-in-the-Middle attack, with extra functionalities. This attack was totally put down by the IPS system.
  - Attack 6 (Infectious Media Generator):
    This attack produces a infectious media file (of the formats .mp4, .jpg, .bmp, .mp3, etc) which is basically embedded with reverse TCP payload which completes the communication back to the attacker successfully. IPS restricted the reverse communication, and removed the payload.
  - Attack 7 (Custom Virus):
    A custom virus which is a folder bomb, was built and executed. It was successful in slowing down the system. IPS had no effect on this virus, the virus continued to function without harm.

- Intrusion Detection System: Zone Alarm
  - Attack 1 (Zenmap):
    The attack simply retrieved the OS details of the host system. The IDS system detected the breach and notified to the user, hence the attack was totally beaten.
  - Attack 2 (Spear Phishing):

This attack generates a payload, a mass mailer attack, which was attached with a reverse TCP payload. The IDS system detected the payload and was successfully able to remove it.

- Attack 3 (Backdoor (Simple)):
  This attack generates a payload, a reverse TCP payload, which successfully causes a reverse connection back to the attacker at all times. The IDS system detected the backdoor payload and removed it successfully, but heavily crashed during its functioning, because it also tried to resolve trivial processes, which was unnecessary.
- Attack 4 (Backdoor (Meterpreter)):
  This attack generates a payload, a reverse TCP payload, which successfully causes a reverse connection back to the attacker at all times. The IDS system detected the backdoor payload and removed it successfully, but heavily crashed during its functioning, because it also tried to resolve trivial processes, which was unnecessary.
- Attack 5 (EvilGrade):
  EvilGrade is a type of Man-in-the-Middle attack, with extra functionalities. This attack was totally put down by the IDS system.
- Attack 6 (Infectious Media Generator):
  This attack produces a infectious media file (of the formats .mp4, .jpg, .bmp, .mp3, etc) which is basically embedded with reverse TCP payload which completes the communication back to the attacker successfully. IDS detected and restricted the reverse communication, and removed the payload.
- Attack 7 (Custom Virus):
  A custom virus which is a folder bomb, was built and executed. It was successful in slowing down the system. IDS detected the issue, but succeeded in only terminating the process and not removing the virus.

- OS Query: OS Query
  - Attack 1 (Zenmap):
    The attack simply retrieved the OS details of the host system. The OSQuery defence system had no defensive effect against the attack, it simply succeeded in displaying open ports and running processes.
  - Attack 2 (Spear Phishing):
    This attack generates a payload, a mass mailer attack, which was attached with a reverse TCP payload. The OSQuery defence system had no effect on this attack at all.

  - Attack 3 (Backdoor (Simple)):
    This attack generates a payload, a reverse TCP payload, which successfully causes a reverse connection back to the attacker at all times. The OSQuery defence system had no effect on this attack at all. The only procedure possible was to kill the processes.
  - Attack 4 (Backdoor (Meterpreter)):
    This attack generates a payload, a reverse TCP payload, which successfully causes a reverse connection back to the attacker at all times. The OSQuery defence system had no effect on this attack at all. The only procedure possible was to kill the processes.
  - Attack 5 (EvilGrade):
    EvilGrade is a type of Man-in-the-Middle attack, with extra functionalities. The OSQuery defence system had no effect on this attack at all.
  - Attack 6 (Infectious Media Generator):
    This attack produces a infectious media file (of the formats .mp4, .jpg, .bmp, .mp3, etc) which is basically embedded with reverse TCP payload which completes the communication back to the attacker successfully. The OSQuery defence system had no effect on this attack at all.
  - Attack 7 (Custom Virus):
    A custom virus which is a folder bomb, was built and executed. The OSQuery defence system helped in killing off this process effectively.

- Disk Encryption: Vera Crypt
  - Attack 1 (Zenmap):
    The attack simply retrieved the OS details of the host system. The Disk Encryption defence system had no defensive effect against the attack.
  - Attack 2 (Spear Phishing):
    This attack generates a payload, a mass mailer attack, which was attached with a reverse TCP payload. The Disk Encryption defence system had no defensive effect against the attack.
  - Attack 3 (Backdoor (Simple)):
    This attack generates a payload, a reverse TCP payload, which successfully causes a reverse connection back to the attacker at all times. The Disk Encryption defence system was used to encrypt a custom drive, which was not at all accessible by the attacker, although the payload still ran undetected.

  - Attack 4 (Backdoor (Meterpreter)):
    This attack generates a payload, a reverse TCP payload, which successfully causes a reverse connection back to the attacker at all times. The Disk Encryption defence system was used to encrypt a custom drive, which was not at all accessible by the attacker, although the payload still ran undetected.
  - Attack 5 (EvilGrade):
    EvilGrade is a type of Man-in-the-Middle attack, with extra functionalities. The attack success successfully attempted the breach but was not successful in accessing the encrypted disk.
  - Attack 6 (Infectious Media Generator):
    This attack produces a infectious media file (of the formats .mp4, .jpg, .bmp, .mp3, etc) which is basically embedded with reverse TCP payload which completes the communication back to the attacker successfully. The Disk Encryption system had no effect on this attack at all, except that the reverse TCP didn't work in encrypted space.
  - Attack 7 (Custom Virus):
    A custom virus which is a folder bomb, was built and executed. The Disk Encryption system had no effect on this attack at all, except that the virus wouldn't work in encrypted space.

- Reputation System: Comodo Security
  - Attack 1 (Zenmap):
    The attack simply retrieved the OS details of the host system. The Reputation system had no defensive effect against the attack.
  - Attack 2 (Spear Phishing):
    This attack generates a payload, a mass mailer attack, which was attached with a reverse TCP payload. The Reputation system had no defensive effect against the attack.
  - Attack 3 (Backdoor (Simple)):
    This attack generates a payload, a reverse TCP payload, which successfully causes a reverse connection back to the attacker at all times. The Reputation system detected the execution of the process and listed it, but didn't cause any action.
  - Attack 4 (Backdoor (Meterpreter)):
    This attack generates a payload, a reverse TCP payload, which successfully causes a reverse connection back to the attacker at all times. The Reputation system detected the execution of the process and listed

it, but didn't cause any action.
  - Attack 5 (EvilGrade): EvilGrade is a type of Man-in-the-Middle attack, with extra functionalities. The Reputation system had no effect on this attack at all.
  - Attack 6 (Infectious Media Generator):
    This attack produces a infectious media file (of the formats .mp4, .jpg, .bmp, .mp3, etc) which is basically embedded with reverse TCP payload which completes the communication back to the attacker successfully. The Reputation system detected the execution of the process and listed it, but didn't cause any action.
  - Attack 7 (Custom Virus):
    A custom virus which is a folder bomb, was built and executed. The Reputation system detected the execution of the process and listed it, but didn't cause any action.

- Automated Response And Remediation: Suricata
  - Attack 1 (Zenmap):
    The attack simply retrieved the OS details of the host system. The Reputation system had no defensive effect against the attack.
  - Attack 2 (Spear Phishing):
    This attack generates a payload, a mass mailer attack, which was attached with a reverse TCP payload. The ARR system detected the breach (only upon intense scan) and successfully stopped its execution.
  - Attack 3 (Backdoor (Simple)):
    This attack generates a payload, a reverse TCP payload, which successfully causes a reverse connection back to the attacker at all times. The ARR system dtected the execution of the process, but didn't succeed in removing it.
  - Attack 4 (Backdoor (Meterpreter)):
    This attack generates a payload, a reverse TCP payload, which successfully causes a reverse connection back to the attacker at all times. The ARR system detected the execution of the process, but didn't succeed in removing it.
  - Attack 5 (EvilGrade):
    EvilGrade is a type of Man-in-the-Middle attack, with extra functionalities. This defence system had no effect on this attack at all.
  - Attack 6 (Infectious Media Generator):

This attack produces a infectious media file (of the formats .mp4, .jpg, .bmp, .mp3, etc) which is basically embedded with reverse TCP payload which completes the communication back to the attacker successfully. The ARR system dtected the execution of the process, but didn't succeed in removing it.

- Attack 7 (Custom Virus):
  A custom virus which is a folder bomb, was built and executed. The ARR system dtected the execution of the process, but didn't succeed in removing it.

- Backup And Rollback
  - Attack 1 (Zenmap):
    The attack simply retrieved the OS details of the host system. The Backup And Rollback system had no defensive effect against the attack.
  - Attack 2 (Spear Phishing):
    This attack generates a payload, a mass mailer attack, which was attached with a reverse TCP payload. The Backup And Rollback system had no defensive effect against the attack.
  - Attack 3 (Backdoor (Simple)):
    This attack generates a payload, a reverse TCP payload, which successfully causes a reverse connection back to the attacker at all times. The Backup And Rollback system had no defensive effect against the attack, but could help revert in case the attack causes a lot of system changes.
  - Attack 4 (Backdoor (Meterpreter)):
    This attack generates a payload, a reverse TCP payload, which successfully causes a reverse connection back to the attacker at all times. The Backup And Rollback system had no defensive effect against the attack, but could help revert in case the attack causes a lot of system changes.
  - Attack 5 (EvilGrade):
    EvilGrade is a type of Man-in-the-Middle attack, with extra functionalities. This defence system had no effect on this attack at all.
  - Attack 6 (Infectious Media Generator):
    This attack produces a infectious media file (of the formats .mp4, .jpg, .bmp, .mp3, etc) which is basically embedded with reverse TCP payload which completes the communication back to the attacker suc-

cessfully. The Backup And Rollback system had no defensive effect against the attack, but could help revert in case the attack causes a lot of system changes.

- Attack 7 (Custom Virus):
  A custom virus which is a folder bomb, was built and executed. The Backup And Rollback system had no defensive effect against the attack, but could help revert in case the attack causes a lot of system changes.

Effective Set Of Defensive Strategies which worked out:

- Disk Encryption:
  No matter which attack strategy is deployed, this defensive strategy has the same role, encrypted a selected disk. All the attack strategies failed to access the data inside the disk, although the breach was successful. So all the attacks were rendered useless. What caused an issue with this strategy is the amount of time it took to get the data encrypted. The system also experienced a lag.
- Intrusion Detection System:
  This strategy worked against all the attacks except the 7th (Folder Bomb). It was successful at detected all the attack payloads and stop the attack. The issue with this system is the amount of time it took to detect the attack.
- Intrusion Prevention System:
  This strategy worked against all the attacks except the 7th (Folder Bomb). It was successful at detected all the attack payloads and stop the attack. Against the 4th attack however (Backdoor using Meterpreter), the system experienced a hang, and the attack was still functional. Hence the issue with the attack was the lag.

The plan is to implement a set of modules which support Backup and Rollback System, Disk Encryption, IDS and IPS system.

Chapter 8

Conclusion

8.1 Conclusion

The objective of this project is to define a set of minimal defensive strategies which work well for the most basic forms of attacks. Although the defensive strategies are available, only a few of them are open-sourced. Also the proprietary ones tend to perform futile tasks and hence consume large amount of computer resources.
The proposed set of minimal defensive strategies succeeds in preventing most of the basic attacks frequently and popularly performed on a normal client system. These defensive strategies are deduced by quantising the hardware parameters before attack, after attack and before defence, and after defence, Lastly, the defensive strategies are successfully implemented in Java, a multi-utility language.

8.2 Future Scope

The proposed defensive strategies were based on the idea that these result in being lightweight on the operating system. From the observations made, optimising security operations compared to existing technologies is possible. Therefore additional work can be done by this application if it is a successful part of an operating system.

Bibliography

[1] Computer Network Attack and Defense Technology - Xinyue Yang,Shujun Zhou,Guanghui Ren,Yaling Liu - Computer Engineering, North Sea University of Technology, Guangxi, China

[2] Cyber Attacks and Defense Strategies in India: An Empirical Assessment of Banking Sector - Atul Bamrara (HNB Garhwal University, India), Gajendra Singh (Doon University, India), Mamta Bhatt (HNB Garhwal University, India)

[3] Network Attack Detection and Defense – Security Challenges and Opportunities of Software-Defined Networking - Edited by Marc C. Dacier, Sven Dietrich, Frank Kargl, Hartmut König (1 - QCRI – Doha, QA, mdacier@qf.org.qa; 2 - City University of New York, US, spock@ieee.org; 3 - Universität Ulm, DE, frank.kargl@uni-ulm.de; 4 - BTU Cottbus, DE, hartmut.koenig@b-tu.de)

[4] CYBER DEFENSE TECHNOLOGY NETWORKING AND EVALUATION - By Members of the DETER and EMIST Projects

[5] Analysis of Network Attack and Defense Strategies Based on Pareto Optimum - Yang Sun, Wei Xiong, Zhonghua Yao, Krishna Moniz and Ahmed Zahir

[6] Online Banking: Information Security vs Hackers Research Paper - Paul Jeffery Marshall - International Journal of Scientific & Engineering Research, Volume 1, Issue 1, October-2010, ISSN 2229-5518

[7] Security Engineering: A Guide to Building Dependable Distributed Systems - Chapter 18 - Network Attack and Defense

[8] The Kali Linux Documentation - The most advanced penetration testing distribution ever.

[9] What is Phishing And How This Cyber Attack Works And How To Prevent It? - https://www.csoonline.com/article/2117843/phishing/what-is-phishing-how-this-cyber-attack-works-and-how-to-prevent-it.html

[10] Spear Phising, Kaspersky Resource Center - https://www.kaspersky.co.in/resource-center/definitions/spear-phishing
[11] Spear Phishing - What is it? - https://digitalguardian.com/blog/what-is-spear-phishing-defining-and-differenciating-spear-phishing-and-phishing
[12] SEToolkit for Kali Linux - https://tools.kali.org/information-gathering/set
[13] Understanding Computer Attack And Defense Techniques - https://zeltser.com/computer-attacks-defenses/
[14] Social Engineering - https://www.incapsula.com/web-application-security/social-engineering-attack.html
[15] Udemy - End Point Protection (Video Content) https://www.udemy.com/the-complete-cyber-security-course-end-point-protection/
[16] Learning Lynda - Computer Security Investigation And Response (Video Content)
[17] Learning Lynda - Learning Kali Linux - June 2017 (Video Content)
[18] India's Cyber Strategy - The Best Defense Is A Good Offense.
[19] Hansman, S. & Hunt, R. (2005). A taxonomy of network and computer attacks, Computers & Security, 24, 31-43
[20] CEH V10 EC-Council Certified Ethical Hacker - Most Demanding Complete Hacking87 Guide - www.ipspecialist.net
[21] Metasploit Toolkit for penetration testing, exploit development, and vulnerability research - www.syngress.com
[22] SECRETS OF A SUPER HACKER - By The Knightmare - Chapter Three - Researching The Hack
[23] Sample Chain Of Custody Form