

CUSTOMIZING CUCKOO SANDBOX'S MALWARE ANALYSIS
REPORTING MODULE

MOHD AIMAN AFNAN BIN MOHD YUSOFF



This report is submitted in partial fulfillment of the requirements for the Bachelor of
Computer Science (Computer Network)

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2016

DECLARATION

I hereby declare that this project report entitled
**CUSTOMIZING CUCKOO SANDBOX'S MALWARE ANALYSIS
REPORTING MODULE**
is written by me and is my own effort and that no part has been plagiarized
without citations.



STUDENT :


(MOHD AIMAN AFNAN BIN MOHD YUSOFF)

Date:

25/8/2016

اونيورسيتي تیکنیکل ملیسیا ملاک

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

I hereby declare that I have read this project report and found
this project is sufficient in term of the scope and quality for the award of
Bachelor of Computer Science (Computer Networking) With Honours.

SUPERVISOR :


(ASSOC. PROF. DR. MOHD FAIZAL BIN ABDOLLAH)

Date:

26/8/2016

BORANG PENGESAHAN STATUS TESIS*

JUDUL : Customization of Cuckoo Sandbox's Malware Analysis Reporting Module

SESI PENGAJIAN: 2015/2016

Saya MOHD AIMAN AFNAN BIN MOHD YUSOFF
(HURUF BESAR)

mengaku membenarkan tesis (PSM/Sarjana/Doktor Falsafah) ini disimpan di Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dengan syarat-syarat kegunaan seperti berikut:

1. Tesis dan projek adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. ** Sila tandakan (/)

SULIT

(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

TERHAD

(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

TIDAK TERHAD


(TANDATANGAN PENULIS)

Alamat Tetap: 80, JLN MUTIARA 3,
TMN MUTIARA 1, 27600, RAUB,
PAHANG

Tarikh: 25/8/16


(TANDATANGAN PENYELIA)

Assoc. Prof. Dr. Mohd Faizal Bin Abdollah
(Nama Penyelia)

Tarikh: 26/8/16

CATATAN: * Tesis dimaksudkan sebagai Laporan Akhir Projek Sarjana Muda (PSM)
** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa.

DEDICATION

This thesis is dedicated to my parents Haji Mohd Yusoff Bin Othman and Hajah Lelawati Binti Haji Mahmud. Also to all Muslims throughout the world, which is in desperate need for an idol to bring back Islamic Golden Age.

“Indeed, with hardships will be ease” (QS 94:6)



ACKNOWLEDGEMENTS

First and foremost I would like express my gratitude to the Lord Almighty, whom without His guidance, for keeping me in the path of righteousness I would not have been able to be as I am today. I would also like to express an endless words of appreciation to Prophet Muhammad (pbuh) for his teachings to be a meaningful human and as a servant of God. Next, I would like to express millions of thank you to my supervisor Assoc. Prof. Dr. Mohd Faizal Bin Abdollah for guiding me throughout this project, thanks for being an open book and the inspiration for me to keep on going. Also, not forgetting my coding masters Umar Mukhtar Bin Hambaran, Rudy Fadhlee Bin Mohd Dollah, Nur Hafizah Binti Mohd Zaki and Nursyafizila Binti Azizan, thank you for your time in teaching me on how to code using the programming language that is used in this project. And special thanks to my friends, for always being there to give me a hand whenever I need it the most.

ABSTRACT

With the prevalence of ubiquitous computing, threats are growing larger regarding the security of our devices which is connected to the Internet constantly. Therefore, there are needs for us to study the behavior of these threats and their attack pattern in order to combat them. Cuckoo Sandbox is one of the tool that using the static and dynamic malware analysis to analyze the behavior and the attack patterns of a malware. This project will embark upon customizing Cuckoo's malware analysis web reporting module, making it more informative and easily understood by users.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

ABSTRAK

Dengan kemunculan perkomputeran sejajar, ancaman siber telah meningkat kerana setiap alat elektronik yang dimiliki oleh pengguna sentiasa mempunyai jaringan kepada Internet. Ini membuatkan setiap alat ini sentiasa terdedah kepada bahaya daripada malware. Jadi, untuk mengatasi masalah ini adalah menjadi satu keperluan untuk mewujudkan suatu perisian yang mampu mengkaji dan mengeluarkan corak serangan daripada malware, supaya serangan-serangan ini dapat dipatahkan. Cuckoo Sandbox merupakan salah satu perisian untuk mengkaji pergerakan malware ketika ia sedang berjalan. Cuckoo menggunakan teknik analisis statik dan dinamik untuk mengkaji malware. Projek ini berkisar tentang menjalankan modifikasi terhadap modul laporan antaramuka pelayar Internet Cuckoo Sandbox. Ini adalah bertujuan untuk memudahkan pengguna memahami laporan analisis malware yang telah dikeluarkan oleh Cuckoo.

TABLE OF CONTENTS

DECLARATION	I
DEDICATION	II
ACKNOWLEDGEMENTS	III
ABSTRACT	IV
ABSTRAK	V
TABLE OF CONTENTS	VI
TABLE OF FIGURES	X
LIST OF TABLES	XII
CHAPTER I INTRODUCTION	1
1.1 Introduction	1
1.2 Research Problem	2
1.3 Research Questions	3
1.4 Research Objectives	4
1.5 Project Scope	5
1.6 Expected Output	5
1.7 Research Contribution	5
1.8 Report Organization	5
1.9 Conclusion	6
CHAPTER II LITERATURE REVIEW	7
2.1 Introduction	7
2.2 Literature Review	8

2.2.1	Domain	8
2.2.2	Keyword	8
2.3	Related Work	12
2.3.1	Malware Analysis on the Cloud: Increased Performance, Reliability, and Flexibility	12
2.3.2	An Efficient Trojan Horse Classification (ETC)	14
2.3.3	Efficient Methods to Trigger Adversarial Behaviors from Malware during Virtual Execution in Sandbox	16
2.4	Facts and Findings	17
2.4.1	Introduction to Malware	18
2.4.2	Evolution of Malware	19
2.4.3	Technologies to Detect Malware	21
2.4.4	Modules of a Sandbox	28
2.4.5	Workings of Cuckoo Sandbox	34
2.5	Analysis of Current Problem	35
2.6	Proposed Solution	35
2.7	Conclusion	36
CHAPTER III	METHODOLOGY	37
3.1	Introduction	37
3.2	Project Methodology	38
3.2.1	Phase I: Literature review	38
3.2.2	Phase II: Requirement analysis	39
3.2.3	Phase III: Project Design	40
3.2.4	Phase IV: Implementation	40
3.2.5	Phase V: Testing and evaluation	40
3.3	Project Milestone and Gantt chart	41
3.4	Conclusion	44
CHAPTER IV	DESIGN	45
4.1	Introduction	45
4.2	Project Requirement	46
4.2.1	Hardware Requirement	46

4.2.2	Software Requirement	47
4.3	Problem Analysis	51
4.3.1	Architecture of Cuckoo Sandbox Reporting	51
4.3.2	Flow Chart	52
4.3.3	Functional and Non-Functional Requirement	54
4.3.4	Data Flow Diagram	55
4.3.5	Proposed Design	57
4.4	Conclusion	62
CHAPTER V	IMPLEMENTATION	63
5.1	Introduction	63
5.2	Software Development Environment Setup	63
5.3	Software Configuration Management	64
5.3.1	Configuration Environment Setup	65
5.3.2	Version Control Procedure	68
5.4	Implementation Status	68
5.5	Conclusion	69
CHAPTER VI	TESTING AND ANALYSIS	70
6.1	Introduction	70
6.2	Test Plan	71
6.2.1	Test Organization	71
6.2.2	Test Environment	71
6.2.3	Test Schedule	71
6.3	Test Strategy	72
6.4	Test Design	74
6.5	Test Result and Analysis	75
6.6	Conclusion	101

CHAPTER VII	PROJECT CONCLUSION	102
7.1	Introduction	102
7.2	Project Summarization	102
7.3	Project Contribution	103
7.4	Project Limitation	103
7.5	Future Works	103
7.6	Conclusion	104
REFERENCES		105
APPENDICES		107



TABLE OF FIGURES

Figure 2. 1: The general analysis flow of this project (M. Schweiger et al 2013)	13
Figure 2. 2: ETC Controlled Library Architecture (Abuzaid et al. 2013).	14
Figure 2. 3: ETC Cuckoo Sandbox Environment (Abuzaid et al. 2013).	15
Figure 2. 4: Structure of Chapter II	17
Figure 2. 5: How Signature Based Works (Shevchenko 2007)	21
Figure 2. 6: Emulation Technology Framework (Boley 2014)	22
Figure 2. 7: Virtualization Framework (Boley 2014)	23
Figure 2. 8: Windows API Framework (Antoniewicz 2013)	26
Figure 2. 9: Steps to Perform DLL Injection (Antoniewicz 2013)	27
Figure 3. 1: Waterfall Methodology	38
Figure 3. 2: Gantt Chart of Project	41
Figure 4. 1: Appearance of HP ProLiant DL160 G5	46
Figure 4. 2: Cuckoo Sandbox Web Malware Analysis Report	48
Figure 4. 3: Oracle Virtualbox Appearance	49
Figure 4. 4: Architecture of Cuckoo Sandbox	51
Figure 4. 5: Flow Chart of Cuckoo Sandbox	53
Figure 4. 6: Context Diagram of Cuckoo Sandbox	55
Figure 4. 7: DFD Level 1 of Cuckoo Sandbox	56
Figure 4. 8: Proposed Design Malware Type Page	57
Figure 4. 9: Proposed Design of Malware Description Page	58
Figure 4. 10: Proposed Design of Analysis Reporting Page	58
Figure 4. 11: Proposed Design of File Details Reporting Page	59
Figure 4. 12: Proposed Design of Signature Reporting Page	59
Figure 4. 13: Proposed Design of Hosts & Domains Reporting Page	60
Figure 4. 14: Proposed Design of Summary Reporting Page	60
Figure 4. 15: Proposed Design of Files Summary Reporting Page	61
Figure 4. 16: Proposed Design of Registry Summary Reporting Page	61
Figure 4. 17: Proposed Design of Mutexes Summary Reporting Page	62
Figure 5. 1: Architecture of Cuckoo Environment	64
Figure 5. 2: Flow Chart of Cuckoo Development	65
Figure 5. 3: Flow Chart of Malware Sample Submission	66
Figure 5. 4: Flow Chart of Customizing the Reporting Template	67
Figure 6. 1: Top-down Testing Strategy (Weißleder 2013)	72
Figure 6. 2: Black Box Testing (Khan 2012)	73
Figure 6. 3: Flow Chart of System Testing	74
Figure 6. 4: Testing Execution of Cuckoo	75
Figure 6. 5: Testing Submission of Malware Sample	76
Figure 6. 6: Testing Original Malware Reporting (File Details) of keygen.exe	77

Figure 6. 7: Testing Original Malware Reporting (Signatures, Screenshots, Network Analysis & Static Analysis) of keygen.exe	78
Figure 6. 8: Testing Original Malware Reporting (Dropped Files, Processes and Volatility) of keygen.exe	79
Figure 6. 9: Testing Original Malware Reporting (Info and File Details) of VirusShare.exe	80
Figure 6. 10: Testing Original Malware Reporting (Signatures, Screenshots, Static Analysis, Dropped Files, Network Analysis, Processes and Volatility) of VirusShare.exe	81
Figure 6. 11: Testing Original Malware Reporting (Static Analysis) of VirusShare.exe	82
Figure 6. 12: Testing Customized Malware Reporting (Info of Submitted File) of keygen.exe	83
Figure 6. 13: Testing Customized Malware Reporting (File Details) of keygen.exe	84
Figure 6. 14: Testing Customized Malware Reporting (Signatures) of keygen.exe	85
Figure 6. 15: Testing Customized Malware Reporting (Screenshots) of keygen.exe	86
Figure 6. 16: Testing Customized Malware Reporting (Network Analysis) of keygen.exe	87
Figure 6. 17: Testing Customized Malware Reporting (Static Analysis) of keygen.exe	88
Figure 6. 18: Testing Customized Malware Reporting (Dropped) of keygen.exe	89
Figure 6. 19: Testing Customized Malware Reporting (Volatility) of keygen.exe	91
Figure 6. 20: Testing Customized Malware Reporting (Info) of VirusShare.exe	92
Figure 6. 21: Testing Customized Malware Reporting (File Details) of VirusShare.exe	93
Figure 6. 22: Testing Customized Malware Reporting (Signatures) of VirusShare.exe	94
Figure 6. 23: Testing Customized Malware Reporting (Screenshots) of VirusShare.exe	95
Figure 6. 24: Testing Customized Malware Reporting (Network Analysis) of VirusShare.exe	96
Figure 6. 25: Testing Customized Malware Reporting (Static Analysis) of VirusShare.exe	97
Figure 6. 26: Testing Customized Malware Reporting (Dropped) of VirusShare.exe	98
Figure 6. 27: Testing Customized Malware Reporting (Processes) of VirusShare.exe	99
Figure 6. 28: Testing Customized Malware Reporting (Volatility) of VirusShare.exe	100

LIST OF TABLES

Table 1. 1: Research Problem	3
Table 1. 2: Research Question	3
Table 1. 3: Research Objectives	4
Table 2. 1: Type of Report	10
Table 2. 2: Type of Malware	18
Table 3. 1: Specification of Server	39
Table 3. 2: Project Milestone	42
Table 4. 1: Server Specifications	47
Table 5. 1: Implementation Status	68



CHAPTER I



INTRODUCTION



1.1 Introduction

With the prevalence of ubiquitous computing, threats are growing larger regarding the security of our devices which almost all of it; from smartphones to personal computers are connected to the Internet. With the continuous connection of our personal devices to the Internet, the larger the security threats has become as almost everything can be done through it; from stealing a user's personal information to planting a bot in one's PC or smartphone.

According to a research conducted by Symantec in their 2015 Internet Security Report, the overall global average hot life span of bot-infected computers in 2014 are 7.5 days, which are slightly higher than 2013, when it was only six days. Meanwhile in the same year, the statistics of mobile threats are recorded that; on Android OS there are 94% of threats, followed by iOS with 6% respectively, then comes Windows and Symbian which is accounted at 0% of mobile threats.

With the rise of computer security threats by year, there is in need of technologies to be used to analyze the workings of the malware on all platforms in order to combat them; from the attack pattern to the registry modified by the malware. Sandboxing technique is one of the most used mechanism to analyze the workings of a malware. In a nutshell, a sandbox isolates malicious programs, untested code, untrusted user or untrusted website to an isolated environment. Thus preventing them to gain critical information or damaging the rest of the computer, it restricts what a program does by just giving it permissions as it needed without adding any other additional permissions that could be abused.

This project will embark on Cuckoo Sandbox to perform an automated malware analysis and to customize its reporting module. The expected output of this project is to understand on how to customize the reporting module of Cuckoo Sandbox. Other than that, this project will also provide a technique on how to customize Cuckoo Sandbox's reporting module. Thus, making it more user-friendly and easier to be understood by new computer security analyst. It is also expected by the end of this project that a technique on how to conduct an automated malware analysis will be documented to ease the future computer security analyst in conducting malware analysis using Cuckoo Sandbox.



1.2 Research Problem

Sandbox is one of the most used mechanism by computer security analyst to study the workings of a malware. Despite that, the GUI of most of the sandbox offered are not very informative and user friendly. Prior to that, this research will focus on how to customize the reporting module of Cuckoo Sandbox and how to perform an automated malware analysis using Cuckoo Sandbox. The main purpose of this research is to contribute to public on how to customize the reporting module of Cuckoo Sandbox. Table 1. 1 below shows the research problem that this project will embark upon.

Table 1. 1: Research Problem

No	Research Problem
1	Difficulties in customizing the reporting module of Cuckoo Sandbox.
2	Lack of customization technique released to public in customizing the environment of Cuckoo Sandbox

1.3 Research Questions

In fact, what is modules in a sandbox? A preliminary study of the sandbox architecture must be conducted before any customization work will take place. Once the architecture is understood then how can the module be customized and how to perform an automated malware analysis using a sandbox? Table 1. 2 below shows the research question that this project will embark upon.

Table 1. 2: Research Question

RQ	Research Questions
RQ1	What is the architecture of a sandbox?
RQ2	How to conduct a customization on a sandbox modules?
RQ3	How to perform an automated malware analysis using a sandbox?

1.4 Research Objectives

Based on the research questions, this project will carry three main objectives. To ensure the project will carry on smoothly. The architecture and the workings of a sandbox will be studied and understood. After the preliminary study is completed, the customization of the sandbox's GUI will be carried out by modifying its reporting module. Finally, an automated malware analysis will be carried out to test the new GUI and to observe the new GUI and reporting module after being customized. Table 1. 3 below shows the research objectives that this project will based upon.

Table 1. 3: Research Objectives

RQ	RO	Research Objectives
RQ1	RO1	To study the process and architecture of a sandbox.
RQ2	RO2	To develop GUI customization of the reporting module for the sandbox.
RQ3	RO3	To perform an automated malware analysis using the sandbox and test the new and improved GUI

RO 1: To study the process and the architecture of a sandbox

An in-depth research will be conducted to fully understand the workings of a sandbox.

RO 2: To develop GUI customization of the reporting module for the sandbox.

The reporting module of the sandbox will be customized to improve its GUI to become more informative and user-friendly.

RO3: To perform an automated malware analysis using the sandbox and test the new and improved GUI.

1.5 Project Scope

Scope of the project is going to be handled as follow:

- i. In-depth research of Cuckoo Sandbox's architecture.
- ii. Focusing on the reporting module of Cuckoo Sandbox.
- iii. Focusing on customizing the reporting module of Cuckoo Sandbox.
- iv. Test the customization by conducting an automated malware analysis.

1.6 Expected Output

The GUI of the chosen sandbox which is Cuckoo Sandbox will be improved. Thus, it will help a lot of computer security analyst in the future.

1.7 Research Contribution

With the rising of computer security threats by year. There is in need of computer security analyst to conduct a malware behavior analysis on the malware that is found on the wild. Prior to the releasing of techniques on how to customize a reporting module of Cuckoo Sandbox and how to perform an automated malware analysis it will sure help the computer security analyst in easing their tasks. And any malware behavior and attack pattern can be understood easily.

1.8 Report Organization

Chapter 1: Introduction

This chapter will focus on introduction, project background, research problem research question, research objective, scope, project significant and report organization.

Chapter 2: Literature review

This chapter will thrive more on the explanation and details of this project, supported with reading materials and conference paper. In this section, other related projects will also be included such as architecture of sandbox, static and dynamic analysis, examples of sandbox and others.

Chapter 3: Methodology

This chapter will explain the method that will be used in this project. The method that is used in this project is the waterfall method. This will ease the task for implementing and organizing the project.

Chapter 4: Design and analysis

In this chapter, software and hardware are coordinated to be used in implementing the project. The customization of the module will be conducted.

Chapter 5: Implementation and testing

This chapter will test the new and improved GUI by conducting an automated malware analysis.

Chapter 6: Conclusion

In this chapter, all project summarization, project contribution and project limitation will be explained. All the steps that have been made and that have been developed for this project will be listed briefly. In this last chapter also explain on additional work can be done in future.

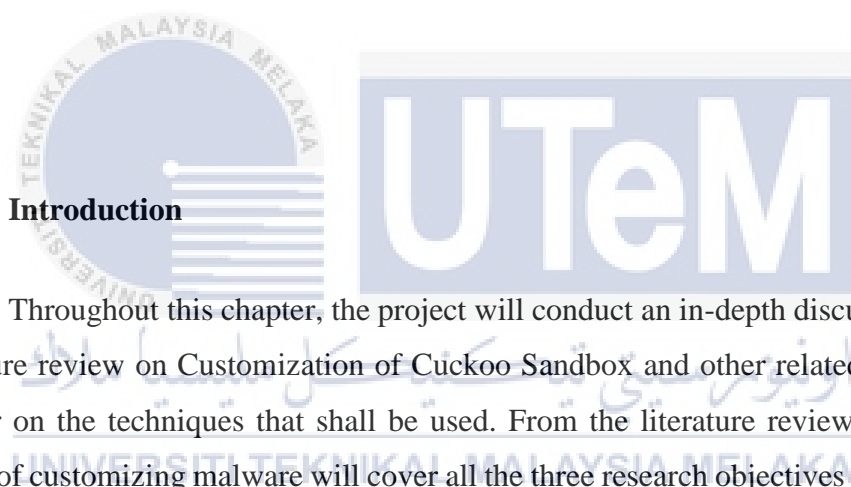
1.9 Conclusion

As for conclusion, at the end of this project, the GUI of Cuckoo Sandbox will be improved and an automated malware analysis will be conducted to test the new GUI. The next chapter will be focusing about literature review. Which will be covering about model approach and related work about sandbox itself and customizing sandbox's module.

CHAPTER II

LITERATURE REVIEW

2.1 Introduction



Throughout this chapter, the project will conduct an in-depth discussion about the literature review on Customization of Cuckoo Sandbox and other related project that is similar on the techniques that shall be used. From the literature review, the results on issues of customizing malware will cover all the three research objectives (RO1, RO2 and RO3), which is to have a clear understanding on the architecture of a sandbox, to conduct GUI customization of a sandbox by modifying its reporting module and to perform an automated malware analysis using the sandbox to test the improved GUI aforementioned in chapter 1.

2.2 Literature Review

2.2.1 Domain

The domain of the project is ICT in Computer Networking focusing on Network Security. Network security has always been an issue that has been dated back for as long as the term computer network first introduced. Since the emergence of the Internet, there has been increasing demands on having a decent security to safeguard their confidential information as it moves across the Internet. One important type of security software that has been introduced to the public is Sandbox. In this chapter, an overview what sandbox is, and introduction to some concepts and theory involved in sandboxing technique will be elaborated.

2.2.2 Keyword

There are several terms that will be used in this project and shall serve the purpose as the keywords throughout the research of the project.

- i. **Sandbox:** Sandbox is an isolated environment initially used by software developers to test new programming code, (Mourad 2015). The implementation of a sandbox is in a virtualized and restricted operating system, by doing this, analyst are able to analyze the behavior of a malware and at the same time controlling the resource that the malware can obtain. Sandbox will conduct dynamic and static malware analysis technique to determine whether a program is considered malicious or not.
- ii. **Types of Sandbox:** Sandbox can be classified into online and standalone sandbox. An online sandbox, as the name has already explain it; are hosted outside of a local environment by an external organization. The usage of online sandboxes enables the public to submit any malicious file through their webpage and it shall serve a

purpose to analyze the file and provide a final reporting of the file behavior for the public. An analyst are not able to customize an online sandbox suit to their needs, as it is fully govern by another organization. Examples of online sandbox solutions are Malwr, Virustotal and others.

A standalone sandbox on the other hand, is fully maintained by the analyst at a local environment. The implementation of a standalone sandbox is to have a dedicated machine which are able to run a virtual machine within it. With standalone sandbox, analyst are able to customize it according to their needs. For example, a testing environment can be created for different versions of Windows. Initially, a standalone sandbox needs to be installed, configured and updated by an analyst to ensure it can work smoothly. Examples of standalone sandbox solutions are Cuckoo, Sandboxie, Buster and others.

- iii. **Modules of Sandbox:** Standalone sandboxed are often written in modular architecture. This is mainly because they wanted the users to be able to customize the sandbox according to their needs. This would then allow the standalone sandbox to be customizable. As for Cuckoo Sandbox, there are six main configuration files (modules) which are cuckoo.conf, auxiliary.conf, <machinery>.conf, memory.conf, processing.conf and reporting.conf. Each modules serves different functionality such as verifying the virtual machines to be associated with the sandbox, what type of reporting to be shown and others.
- iv. **Sandbox Reporting Module:** Sandbox will serve the function of conducting an automated malware analysis of a malicious file that has been uploaded into the system. Once done, the sandbox shall give out a report on the behavior of the file thus informing the analyst if the file is malicious or not. Table 2. 1 below shows the report that will be given by the sandbox, it will include the following information, the reporting however might vary on different sandbox:

Table 2. 1: Type of Report

Type of Report	Explanation
Filename	The name of the file will be shown in the report. E.g. server.exe
Filetype	The filetype of file will be explained in details. E.g. MPEG ADTS, layer III, v2, 16 kbps, 16 kHz, Monaural
MD5	Hashing technique. An algorithm to identify the data integrity. Creates a 128 bit message digest from data input. Unique to specific data. Eg. 9988093c8033f0d636102ef00b5dee6c
SHA1	Hashing technique. An algorithm to identify the data integrity. Creates a 160 bit message digest from data input. Also rendered as hexadecimal number of 40 digits long. Unique to specific data. E.g. 0222bacc7f3592245e6fd79706732c345a77f40d
SHA256	Hashing technique. An algorithm to identify the data integrity. Generates an almost unique 256 bit signature for text. E.g. 51570d595fad9c2d13dd446a0db68e9869cd2fed66619ae57a32a484b2717e79
SHA512	Hashing technique. An algorithm to identify the data integrity. Generates an almost unique 512 bit signature for text. E.g. 5bc5bab229576fec83c644e94c42ba35451d3528866500aeb73b40cee28fa3e4f87e2e6a830ed7de74445cafa2664e70402b1b18e39119c38f7ff39e3ca419b
CRC32	Cyclic Redundancy Code. For error detection. Hash function that is based on polynomial division idea. A long binary number is taken then divided by a constant divisor. Eg. 1D846AA2
YARA	A tool that is to help malware analyst to identify and classify malware samples. Using this tool an analyst could classify a malware family based on text or binary patterns.