Implementation, testing and validation report for

Threat Intelligence Platform (TIP) system

Prepared by : GROUP: BSE 18 – 49

Submitted to: Mr. Alex Mwotil

June 2018

Document No:	Doc_1.0
Prepared by:	Group 49
Date:	Wed, June 13, 2018
Version:	1.0

Document Approval

Name	Role	Date	Signature
SSEKITTO RONALD MASETE NICHOLAS MWESIGWA BILL JOHN KATUMBA ARNOLD	Back-end developer/System tester /engineer Documenter/Publisher /Database designer Architectural designer/test engineer/system analyst/quality analysts/ documenter Front end developer/System tester/project manager/documenter/ quality engineer		
	Validation		
	Client		

Declaration

We, group BSE 18- 49, hereby declare that the work presented is original and has never been submitted to any other university or institution of higher learning for an award. We can confirm that where we have done consultations either from published material or from the works of others, it has been attributed in this report.

	NAME	**REGISTRATION NUMBER**	**SIGNATURE**
1	Masete Nicholas	14/U/13679/EVE	
2	Katumba Arnold	14/U/7399/EVE	
3	Mwesigwa Bill John	14/U/10371/EVE	
4	Ssekitto Ronald	14/U/14824/EVE	

Date: …….

Approval

This project report titled " Threat Intelligence Platform (TIP) system" has been submitted for examination with my approval as the supervisor of group

BSE 18-49.

Signature: .....

Date: ……..

Mr. Alex Mwotil

Project Supervisor

Department of Networks

School of Computing and Informatics Technology

Makerere University

Dedication

We dedicate this work to the Almighty God, our supportive families and friends for all the love, care and sacrifices they have offered to see us through this project. We thank them for all their encouragement and support that has geared us towards achieving our professional goals and made us who we are today.

Acknowledgments

To the Almighty God, we thank him for the love, affection and mercy he had for us throughout the development of our final year project up to the time of producing this report.

Our heartfelt gratitude goes to our supervisors; Mr. Alex Mwotil and Mrs. Mary Nsabagwa; who have been very instrumental in this project by selflessly providing their time, expert advice and guidance that has enabled us complete this project.

The support of family and friends during the period when carrying out this project cannot be left unsaid. Our parents most especially have been so amazing during this period. To all other family members and friends, words cannot fully express our gratitude to you but we sincerely appreciate you for what you have been to us throughout the whole period of our career development.

We are also sincerely grateful to our project team members for their dedication, hard work and cooperation throughout the entire academic year to see this project to completion.

Table of Contents

1 Introduction

1.1 Overview of system

Many organizations fail to identify threats ending up spending most of there time on the wrong areas or spend too long on processes, such as risk and vulnerability analysis, instead of mitigating and fixing issues. Making effective use of Cyber threat intelligence is an important component of an organization's security program. Effective use of Cyber threat intelligence (CTI) is an important tool for defending against malicious actors on the Internet.

The threat intelligence platform is a security system that collects malicious feeds from different sources using the help of a tool known as intelmq, and uses these feeds to making sense out of this data through analysis, and visualization to help organizations understand and manage business risk turning unknown threats into known and mitigate threats, and to improve the effectiveness of defense. TIP provides a solution for IT security teams through collecting and processing security feeds and stores the results in mongo database. These feeds are retrieved from the DB for analysis using python and put back for further use. The clients accesses the analyzed feeds through a defined server component.

1.2 Overview of document

This document contains documentation of the system validation activities. The tables are filled in with information about the tasks performed, methods used, criteria for acceptance, input and output required for each task, required documentation, the persons that are responsible for the validation among others.

Tasks performed	**Methods used**	**Acceptance criteria**	**Input required**	**Output required**
Collection of feeds	Intelmq	Malicious feeds		
Storing of feeds	Mongodb	JSON format	Malicious feeds	feeds
Analysis	Flask framework	JSON format	Malicious feeds	Analyzed malicious data
Client request	Special component		request	Corresponding feedback

2 Requirements and system acceptance test specification

The requirements describe and specify the system completely and are the basis for the development and validation process.

Table 1 Requirements specifications

Topics	**Requirements specification**
Version of requirements	✓ Document version 1.0, of the requirements specifications is what is followed in the sub sequential deliveries of the system. ✓ No major changes to the specification has been done.
Input	✓ Raw feeds collected from external sources ✓ user login data – This includes both the user-name and password. ✓ Incorrect input data is also submitted to the system which is caught by input validations.
Output	✓ Analysis results are output/stored in MONGO database. ✓ The system outputs graphs showing timely results of attacks. ✓ Feedback is drawn back to the user for every request made. ✓ Collected feeds are output to MONGO database. ✓ Requests to and responses from the system are transferred in JSON format. ✓ The system enables view of analysis details via the dashboard. ✓ Clear and meaningful errors messages are displayed by the system. ✓ The system generates logs that can be viewed by the administrators.

Topics	**Requirements specification**
Limitations	✓ It does not support any other data format other than JSON objects. ✓ The system runs only on a Linux operating system. ✓ The system cannot by itself start the intelmq BOTS. ✓ The system cannot block an address from accessing the network, but can only block it from accessing a certain machine. ✓ The system shall work where there is an Internet connection. Since the product shall require fetching data from the database over the Internet, it is crucial that there is an Internet connection for the product to function. ✓ As of present, users of this system have to understand English language as user interface information is displayed in English. ✓ Off-line usage is not enabled. ✓ Users can not view project details unless they are logged onto the system. ✓ Users are required to be Internet literate.
Safety	✓ The system is safe and secure to its users, there is no any danger to the health of the system users or physical harm expected. ✓ For every incorrect input to the system, there is a handler for errors to avoid system malfunctioning
Default settings	✓ The system has some default settings that can be used at the time of installation, they serve as the standard settings, but can be changed later on via the system settings interface. ✓ The system READ_ME provides a description of these settings in details.
Version control	✓ The system versions are defined by numbers and the current pre-release version being TIP-v1. Final releases will be marked by the final phase. (ie TIP-final-1) ✓ The system is versioned by **git** stored both locally on hard disks and remotely on the **gitHub** repository. Subsequent versions are to be identified by the commits made on the system after bug fixes, additional or removal of features, optimality added, and these changes are to be clearly seen from the messages added to commits.

Topics	**Requirements specification**
Dedicated platform	✓ The system works only and best on a Linux operating system. ✓ It is to be installed locally on a network but have access to external tools, meaning it cannot be accessed by external parties unless if its the users will. ✓ The system was not tested on a MAC OS. ✓ Since feeds will be collected continuously, a constant power supply is required. ✓ For the hardware specifications, a minimum dedicated storage of 500GB hard disk, a minimum of 4GB RAM, for the ultimate and real-time performance will be required.
Installation	✓ The system provides an installation disk (CD). The disk will contain both the system and the installation instructions. The instructions will be located in a system READ_ME file.
Service and maintenance	✓ Different versions will be released in an approximately three weeks time for urgent updates and patches, and a fixed release of 3 months. The early release is because security weakness/vulnerability can be exploited in a very short time and if not patched quickly, things might go out of hand. The following are considered for future updates. ✓ The future system shall support other languages as needs arises.
Errors and alarms	✓ No alarms installed or Incorporated in the system. ✓ Errors are handled with both client and server validations, with corresponding error feedback. ✓ Clear and meaningful error messages are displayed to the user in case of any errors. ✓ Anomalies that may arise and not catered for are to be tracked down and solved from system logs.

3 Design and implementation process

3.1 Development plan

This section describes Development tools, manpower, and methods used in developing the TIP system.

- ✓ IntelMQ. For collecting malicious data feeds

- ✓ Angular5 for front end designing and developing.
- ✓ Flask Python framework for back-end developing.
- ✓ MongoDB used to implement the database.
- ✓ Studio 3T, used as the database tool for the system.
- ✓ Json is the format to which our data is being stored.
- ✓ A team of four developers to develop the system.
- ✓ Different text editors were used by different group members at different times and these include; WebStorm and PyCharm.

3.2 Design inputs and outputs

Description of the system modules implemented

The system has an analysis module that receives feeds from an external tools, intelmq in this case. This tool keeps the feeds in the database and act as inputs to the analysis package. The only general out put from analysis is are the results,which are then stored back in the database. It also receives analyzed feeds from the analysis module, the clients requests are also considered inputs. The outputs of the system from such requests are the responses given back to the clients. And these outputs are dependent on the client requests,

The design output must meet the design input requirements, contain or make references to acceptance criteria, and identify those characteristic of the design that are crucial to the safe and proper functioning of the product. The design output should be validated prior to releasing the system for final inspection and testing.

Table 1. Design output checklist

Topics	**Design output**	
Implementation (coding and compilation)	Analyzed and ready to visualize data System Design Document. System final implementation.	
Version identification	Versions will be classified in terms of alpha, beta, release and then final release. For example TIP-v1-alpha, TIP-v1-final.	
Good programming practice	Source code is… ✓ Modularized ✓ Encapsulated ✓ Functionally divided ✓ Fail-safe(handling errors)	Source code contains… ✓ Revision notes ✓ Comments ✓ Meaningful names ✓ Readable source code ✓ Printable source code

Topics	**Design output**
Dynamic testing	✓ All statements have been executed at least once ✓ All functions have been executed at least once ✓ All case segments have been executed at least once. ✓ All loops have been executed to their boundaries. ✓ All parts were subject to dynamic test.
Utilities for validation and testing	✓ Local host services were used to test the system's functionality simulating real on-line network. The system was tested on-line using black box testing in which test data was entered and results verified from the expected out puts. ✓ The logic used on different modules was tested when on local host using white box testing in which several code refinements were made and instantly checked by sample inputs.
Inactive code	None.
Documentation	✓ The software design document is the design output of this section. It specifies how the system components are designed and configured to work together. This software design document describes the architecture and system design of TIP system. This document was used by people with different skill sets like project management, system analysis, testing, system design and programming skills.

3.3 Design changes

No design change

3.3.1 Design change justification

None

3.3.2 Design change evaluation

None

4 Inspection and testing

The inspection and testing of the system is planned and documented in a test plan. The extent of the testing is in compliance with the requirements, the system acceptance test specification, the approach, complexity, risks, and the intended and expected use of the system.

The test plan is created during the development or reverse engineering phase and identify all elements that are about to be tested. The test plan should explicitly describe what to test, what to

expect, and how to do the testing. Subsequently it should be confirmed what was done, what was the result, and if the result was approved.

4.1 Test Objectives and types

Topics	**Test plan and performance**
Sequence of tests	Access Control tests, configuration tests, calculation tests, regression tests, Data tests
Data Type Tests	Data is tested for JSON format, if found not to be in JSON format it is converted to. Tokens generated are different for every different feed.
Configuration tests	The system was tested on a windows platform, and the results were not good since it was developed to run on a Linux system. Its recommended to use a Linux OS. The system is meant to run as a standalone system, but possibly integration with other systems might be an option in future releases.
Calculation tests	Calculation tests were carried out on different modules to confirm that the desired output is got and so it was.
Regression tests	Modules of the system were made as independent as possible, so changes in one module don't need changes in other modules. All tests indicated that this stage was successful.
Action if errors	All possible errors were tested for their presence and those found were handled carefully to avoid system malfunction. But in case of any error not observed by our team, the client can directly get involved with the developers via contacts available on the system interface.

4.2 Test Results

- ✓ An account of the sample user was created.
- ✓ A sample user was successful logged in.
- ✓ On clicking log in, the system displayed an error message after using information that does not exist on the system.
- ✓ Real data captured from by intelmq was used to perform analysis and results were displayed to user.
- ✓ Results were displayed graphically.

5 Installation and system acceptance test

The validation of the installation process ensures that all system elements are properly installed in the host system and that the user obtains a safe and complete installation, especially when installing software products.

Table 1 Installation summary

Topics	**Installation summary**
Installation method	Installation is manual, all details concerning the installation process are available on the installation media.
Installation media	With the current version, the installation can be downloaded from the Internet before the final version is released. The final version will be available on a CD-ROM.
Input files	READ_ME.md file – this includes the installation instructions and system default settings. The system it self. All the rest are system files and are equally important.
Installed files	TIP-System.Zip
Supplementary files	READ_ME.md file
Installed components	On importing the database to the database server, One has to get the username, password, host and database name to put them in config folder to the settings.py file.
Installation qualification	The system is running well on the browser and queries to the database are working fine. And the analysis package is also able to apply blocking and unblocking features.

6 Performance, servicing, maintenance, and phase out

In this phase the system is in use and subject to the requirements for service, maintenance, performance, and support. This phase is where all activities during performance reside and where decisions about changes, upgrades, re-validation, and phase out are made.

Topics	**Performance and maintenance**
Problem / solution .	✓ Forgetting a password. **Solution**: Update/reset password using a link provided at the login page. ✓ Client failing to interpret the graphs after the feeds have been analyzed. **Solution**: We refer the client to the user manual as it contains the procedures of how to use the system. ✓ Failing to either carry out an operation such as having a black page. **Solution**: One is advised to try again. Or to check the Internet connection.
Functional maintenance	✓ When a system is required to support another language on the client side, the language needed will be supported.
Functional expansion and performance improvement	✓ Adding DNS pharming functionality to redirect network traffic.

7 Conclusion

By the subsequent signatures it becomes evident that all validation activities are documented and approved

Appendix:

Appendix A: User manual

Refer to the User Manual and Installation guide

Appendix B: Publication

Threat intelligence platform published on Zogodo.org link: on 6th June 2018 **Final approval for use**	
Identification:	
Responsible for validation:	
Remarks:	
Date:	Signature: