

Princess Sumaya University for Technology

King Hussein School for Computing Sciences

**BRO IDS Log files analyzer and visualizer**

**Prepared By:**

Ahmad da'na  
Musaab abushanab

**Supervised By:**

Dr.Ali Hadi

Project Submitted in partial fulfillment for the degree of Bachelor of Science in Computer Science

Spring-2016

Declaration of Originality

This document has been written entirely by the undersigned team members of the project. The source of every quoted text is clearly cited and there is no ambiguity in where the quoted text begins and ends. The source of any illustration, image or table that is not the work of the team members is also clearly cited. We are aware that using non-original text or material or paraphrasing or modifying it without proper citation is a violation of the university’s regulations and is subject to legal actions.

Names and Signatures of team members:

Ahmad da'na

Musaab abushanab

Acknowledgments

To our parents, our families, loved ones, friends and colleagues :

thank you supporting us to do this project .

To our instructor Dr.Ali Hadi :

thank you giving us the motivation to work hard on this project , we could not have done this without your close and caring supervision

Summary

BRO IDS is a network intrusion detection system that generates log files based on the traffic is getting to the host machine which it is installed to.

Our project deals with the logs files generated by the IDS, analyzes the traffic information that is stored in log files, builds relations, displays time lines and graphs of traffic and provides statistics about the information in log files.

This project is intended to make network security analysis easier.

Give a complete but concise description of your work. The summary is a brief overview of your motivation, statement of purpose, general methodological approach, major results, discussion and conclusion. The abstract should not exceed one page.

List of Abbreviations

**PyQt**: Python binding of the cross-platform GUI toolkit Qt

**IDS:** intrusion detection system

**BRO**: an open source intrusion detection system.

**GUI:** graphical user interface

**SQL**: Standard Query Language

**Sqlite** : files based database management system

**UI**: user interface

**PYUIC**: Python user interface converter

**DBMS** : database management system

**QML**: Qt modeling language (a user interface markup language)

**CASE :** computer aided software engineering tools

PSUT: princess Sumaya university for technology

Table of Contents

The table of contents should be automatically generated by going to: *Insert >> Index And Tables >> Table of Contents.* Choose *Classic* as the format of the table and set the number of levels to be 3.

In order for the table of contents to be generated correctly:

* Each chapter title should be formatted using the style “Chapter”.
* Each section title should be formatted using the style “Section”.
* Each subsection title should be formatted using the style “Subsection”.

The table of contents should replace all of the text that is in this page.

Table of Figures

The table of Figures should be automatically generated by going to: *Insert >> Index And Tables >> Table of Figures.* Choose *Classic* as the format of the table and set the tab leader to dots.

In order for the table of figures to be generated correctly, the label of each figure should be formatted using the style “Figure”.

The table of figures should replace all of the text that is in this page.

**Table of Tables**

The table of tables should be automatically generated by going to: *Insert >> Index And Tables >> Table of Figures.* Choose *Table* as the caption, *Classic* as the format of the table and set the tab leader to dots.

In order for the table of contents to be generated correctly, the label of each table should be formatted using the style “Table”.

The table of tables should replace all of the text that is in this page.

Chapter 1   
Introduction

1.1 Overview

BRO is an open-source software, designed by Berkeley university for network traffic analysis and intrusion detection used for network forensics purpose, and is a fully passive network traffic analysis tool.

The software generates log files that log network interactions, and splits the results based on their application layer protocol (HTTP,FTP,DNS SSL ,SMTP and others).[1]

To understand what the project is, the reader should have solid understanding of network concepts and protocols ,a background in security and intrusion types.

Give a general overview of the project, its importance and why you have chosen to work on it. Provide also a discussion of any scientific/technical background that is required to understand what the project is and the motivation behind it.

You can choose to arrange the information into subsections numbered as 1.1.x. The title of each subsection should the style named “subsection”.

1.2 Problem Statement

The data provided by the IDS log files are often hard to deal with, In example of capturing big amount of traffic packets .

We are building a desktop application that reads BRO-generated log files, analyzes them and generates visualized graphs and statistics based on the data provided by the log files.

The target audience is the BRO IDS users, the application would ease the process of studying the traffic and security analysis of networks.

In this section provide (at least) the following: 1) A precise description of the problem this project tries to solve, 2) A description of the outcomes of this project (example: a mobile app, a desktop application, etc) and 3) A description of the target audience/customers, how they will use the system and what impact it will have on them.

1.3 Related Work

after doing several searches around the Internet , we have no information about any program built to deal with and analyze bro generated log files .

Discuss in detail systems that are similar to your system. Provide a critical evaluation of these systems and explain how your system compares to them. Clearly mention if your system uses ideas/features from these systems.

1.4 Document Outline

Describe how this documentation is structured and what will be discussed in each of the following chapters.

Chapter 2   
Project Plan

2.1 Project Deliverables

List and describe the deliverables of the system. Examples of deliverables include: source code, documentation files, executables, datasets, databases, etc.

2.2 Project Tasks

|  |  |  |
| --- | --- | --- |
| Team member | Ahmad da'na | Musaab abushanab |
| Learn python3 |  | 2 weeks |
| Design GUI | 6 weeks | |
| Loading single files and  Loading files within directories | 3 weeks |  |
| Connect and set up the database |  |  |
| Implementing traffic timeline |  |  |
| Sql query processing and command managing | 1 week |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Subdivide the project into high level tasks and provide a timeline for the completion of each task. Tasks can be: learning tasks (e.g. learn Python), documentation tasks (e.g. prepare a requirements document), technical tasks (e.g. develop a web-service) or any other relevant task (e.g. distribute a survey).

Format the information you provide into a table that includes the following for each task: a task number, a name and description, a time duration, and what dependencies need to be completed before the task begins.

Provide also charts like Gantt Charts and PERT Charts to illustrate the timeline of the project.

2.3 Roles and Responsibilities

Discuss the roles and responsibilities of each team-member in relation to each of the tasks. Be specific and provide enough details to allow an outsider to judge the workload for each team-member.

2.4 Risk Assessment

For each task, describe any associated risks that may prevent completing them. Indicate how probable the risk is, its impact on the system and how you plan to cope with it if it appears.

2.5 Cost Estimation

Discuss any costs that are required for implementing the project. Costs may include buying hardware, software licenses, or even costs of paperwork (for example).

2.6 Project Management Tools

1) bitbucket.org was used as a version control system.

2) Qt Creator (a cross platform GUI builder) was used to build the GUI.

3) pyCharm community edition, python IDE and CASE tool

Chapter 3   
Requirements Specification

3.1 Stakeholders

|  |  |  |
| --- | --- | --- |
| Stakeholder No. | Name and description | Data gathering method |
| 1 | Dr. Ali Hadi – professor at PSUT (a BRO user) | interview |
| 2 | BRO IDS users  (mainly higher studies students at PSUT) | Questionnaires and interviews |
|  |  |  |

A stakeholder is any person or entity that is affected by the system or affects in in any sense. In other words, the stakeholders of the system are any users or entities that have an effect on the system requirements.

Describe each stakeholder, his interaction with the system and the importance of his role. Use tables to simplify the provided information.

3.2 Platform Requirements

the program is cross-platform it can run on machines running windows , FreeBSD ,OpenBSD and mac.

the system can run on windows-based machines despite that BRO IDS does not work on windows based machine, the system can work on BRO log files in any platform.

The system requires python3 to run on any system , with its special modules

Sqlite3 and PyQt5 must be installed on the machine for the program to run.

Specify the *software* and *hardware* requirements for running the system. Clearly mention which requirements are a must and which are only recommended.

For systems that are made of sub-systems (e.g. a client side and a server side), make sure to list the requirements for each sub-system separately. For example, the client may need a browser to use the system, whereas the server may require different (more demanding) software and hardware requirements in order to respond to the client requests.

3.3 Functional Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | Name | Description | Note |
| 1 | Accuracy  (essential) | The program should provide accurate analytical graphs and statistical data using the appropriate statistical methods. | \_\_ |
| 2 | Files  (essential) | The program should be able to deal with BRO single. | \_\_ |
| 3 | Directories  (essential) | The program should be able to deal with directories that BRO have multiple log files. | -- |
| 4 | Files  (essential) | The program as well should be able to detect corrupted or unsupported log files provided by user | -- |
| 5 | Traffic time line  (essential) | The program should provide a timeline for traffic. | -- |
| 6 | Results file  (essential) | The program must generate results file that stores the results of the analysis. | -- |
| 7 | Encryption  (essential) | The generated results must be encrypted. | -- |
| 8 | Database  (essential) | The program should clear the database table each time it starts . | -- |
| 9 | Database  (essential) | The program must clear the data base before loading a new file . | -- |
| 10 | Database  (essential) | Database should be normalized . | -- |
| 11 | Warning messages  (essential) | Show message boxes for warning and exceptions. | -- |
| 12 | database | The program must allow the users to deal with the log files in SQL fashion. | -- |
| 13 | Files (essential) | The program should show warning messages for unsupported files , which paths are provided by the path text entry. | -- |
| 14 | Files (essential) | The program should show warning messages for directories that have no log files. | -- |
| 15 | Files (essential ) | The program must check the structure of the log file before it starts analysis. | -- |
| 16 | Files warnings  (essential) | The program must show warning messages if the file does not have the support log file structure. | -- |
| 17 | Files (essential) | The program must inform the user about the progress of files loading via the progress bar. | -- |
| 18 | Files and directories (essential) | The program must inform the user about the progress of loading files in directory via the progress bar too. | -- |
| 19 | Files warnings (essential) | The program must show a warning message if there is a files / directories already loaded in the database | -- |
| 20 | Files warnings  (essential) | The program must warn the user if the user tried to load a file after a log file is already loaded | -- |
| 21 | GUI  (essential) | To prevent errors and exceptions analysis tabs and plotting tab must be disabled before loading files into database | -- |
| 22 | File contents  (essential) | The program must normalize epoch time format to human readable format | -- |
| 23 | Databases  (essential) | The program must warn the user when there is an error connecting and creating the database files  in example of requiring administrative privileges | -- |
| 24 | SQL commands  (essential) | The program must convert SQL command to lower case  (this will not affect database tables names since SQL is case insensitive ) | -- |
| 25 | SQL commands  (essential) | The program must detect select statement before pressing the “execute command” button | -- |
| 26 | SQL commands  (essentials) | The program must detect insert statement before pressing the “execute command” button | -- |
| 27 | SQL  commands  (essential) | The program must be able to execute multi-line SQL queries | -- |
| 28 | SQL commands (essential) | The program must load the selected records from database into the table view based on the condition provided with the SQL command | -- |
| 29 | SQL  commands  (essential) | The program must warn the user about miswritten SQL commands | -- |
| 30 | SQL commands  (essential) | The program must display details about the error in SQL command for user to make suitable corrections | -- |
| 31 |  |  |  |

Provide a detailed list of all the functional requirements of the system. For each requirement, specify exactly what the input, output, processes and main constraints are. Mark also each requirement as either recommended or essential.

Use a table that contains a numbered list of the requirements and their related information. This will facilitate understanding the requirements and referencing them in the proceeding sections and chapters.

3.4 Non-Functional Requirements

|  |  |  |
| --- | --- | --- |
| Requirement | Name | Description |
| 1 | Usability | The program should provide easy to use interactive UI |
| 2 | Performance and responsiveness | The program should be fluid and efficient in terms of performance and memory |
| 3 | Platform compatibility | The BRO IDS is mainly targeted to LINUX/FreeBSD , therefore the visualizer must support these OS's |
| 4 | Open source | The source code of the project will be provided on GitHub under XYZ license |
| 5 |  |  |

Examples of non-functional requirements include: requirements related to performance, storage limits, code quality, documentation, accessibility, security, reliability, scalability, portability, user interface ease-of-use, etc.

3.5 Other Requirements

Include here any requirements that may not directly fall under any of the sections before. Examples include restrictions on which APIs can be used, data transmission protocols, data storage formats, etc.

Chapter 4   
Software Design

In this chapter, provide both a high level and low level design of the system you will develop. You will follow in your design either the structured approach or the Object Oriented approach. Here is a list of the diagrams you need to provide in either case:

**Structured Approach:**

* Functional Decomposition Diagram (FDD).
* System Context Diagram (Context DFD).
* Data Flow Diagram (DFD-0 and lower level DFDs).
* Data Dictionary.

**Object Oriented Approach:**

* Use Case Diagrams.
* Object Diagrams.
* Package and Class Diagrams.
* Component Diagram.
* Deployment Diagram.
* Activity Diagram.
* Sequence Diagram.
* State Transition Diagram.

In both approaches, you need to describe the design of the data in the system. Use Entity Relationship Diagrams (ERD) and provide the detailed database schema.

Describe also any design choices that are related to the user interface. Describe the different screens (or web-pages), and how the flow moves between them. Use appropriate diagrams like trees to describe the structure of web-pages (for example).

Make sure to organize this chapter into sections and subsections in a manner that is appropriate to the provided information. Make sure also to stick the formatting used in the other chapters.

Chapter 5   
Implementation

the system was developed entirely using python3, we chose python3 regarding the easy syntax it provides

we used Qt creator to develop the UI , how ever the tool can develop a UI and its functions entirely using c++ , c++ was not a good option since it requires more lines of code and does not have that number of modules that python3 provides

Qt creator creates a QML UI .

we used PyQt5 Python3's module to implement the UI .

we used a program that PyQt5 provides called PYUIC , a program that converts Qt creator's generated QML files to Python3 files.

We used Sqlite3 DBMS ,because we are not using a multiple clients or other applications running concurrently to access the DB at the same time while the application is using the database, beside , there is no traffic or requests are going to be handled on the host that would interact with the database.

We used PyCharm IDE , since it delivers auto completion of code and a sophisticated syntax errors correction capabilities which would easy the process of coding beside it is free to use for single user

camelCase coding conventions to name variables

This chapter should describe issues related to the implementation of the system. Make sure to include the following:

* A description of all of the programming languages, tools and APIs used and the reason behind choosing them.
* A general overview of the number of implemented classes, packages, scripts, lines of code, SQL queries, pages, etc.
* A description of any coding conventions used.
* A discussion of the main/critical algorithms developed or used. Describe the reason behind choosing/developing the algorithm, the main idea of the algorithm, and provide a pseudo-code for the algorithm or alternatively show snippets of the code (if the code is clear).
* A listing of which features (mentioned in chapter 2) have been implemented and which have been deferred for future iterations of the project. Use the same table provided in chapter 2 with an extra column to indicate if the feature has been implemented or not.

Subdivide your discussion into sections as appropriate and do not hesitate to discuss aspects of the implementation that are not listed above. Do not include all of the code of your project in this chapter.

Chapter 6   
Testing

6.1 Testing Approach

Discuss the testing techniques you have used for each aspect of the system or group of features in the system.

6.2 Testing Tools

Describe any testing tools you have used.

6.3 Tested Features

we have tested the following feathers

files loading since it's a crucial part of the program

Discuss here which features were tested and which were not and explain why.

6.4 Discussion (optional)

Discuss here any test results that have impacted the design or implementation of the system.

Chapter 7   
Conclusions and Future Work

Summarize any results achieved in this project and discuss how you intend to extend the project in the future.

Appendix A   
Users’ Manual

Provide a description (textual and pictorial) of how the system can be installed and used and how the errors messages should be interpreted.

Appendix B   
Document Changes

Discuss here any changes you have made to the document you have submitted in Project 1 and the reasons behind the changes.

Appendix C   
Code Documentation

List the code documentation. Such a documentation can usually be automatically generated using tools like JavaDoc and Doxygen.

References

[1] :