Detection and Termination Technology (DTT) Requirement Analyses

By: Hassan Nasser Eldeen.
Signature:
Date of submission: 23/2/2022

Instructor: Dr. Mohamad Elhaj
Signature:
Date of approval:

Note:

Contents

Glossary:

- DTT: “Detection and Termination Technology” a security system focused on IOT fog level security.
- Functional requirements: a set of information and rules that a system should follow that are aspects of its functionality, and capabilities.
- Non-functional requirements: a set of information and rules a system should follow which affect its non-functional aspects such as, its look, its philosophical standards (privacy, human rights…)
- Namp: “Network Mapper” a network scanning technology
- IOT: “Internet Of Things” is the connection between machine, devices, appliances, etc.
- Gantt chart: named after Henry Gantt, it is a bar chart specifying the scheduling of a certain project.
- Wireshark: a network packet analyzer.
- Packet: the form in which data travels within a network.

Introduction:

System Introduction:

Regular everyday home appliances, are taking the next step into integration with the ever-growing realm of what is considered “smart”. IOT technologies offer users far reaching benefits, from making life easier and more efficient, to making mundane items more interesting. With all the luxuries IOT technologies offer, a lot is being left behind in terms of security, and privacy, due to the low processing power of IOT devices, and to somewhat carelessness in attitudes towards privacy and information security, IOT devices have been left holding user’s personal information, with no effective way to protect it from malicious actors.

DTT is a system aimed to solve issues concerning IOT security at the “fog” level; it does so by having extensive permissions and authority within the home network. Keeping the client’s information private and secure is the focus of DTT, so itself it aims to be secure against any type of malicious acts, for it plays an enormously sensitive role in a home’s network, with great and far-reaching capabilities. It consists of a computer program called “DTT server”, and a mobile application called “DTT RCI (Remote Control Interface)”.

Report Overview:

This paper is a listing of the functional and none-functional requirements, for the Detection and Termination Technology system. Since DTT comprises of two main sections, DTT server and the mobile application, the requirements of each are sectioned on their own.

Scope:

This paper aims to set for the design, and implementation phases the main functionalities, and borders to operate within. It aims to be a reference for the upcoming stages in development, where goals and milestones are set around the requirements reached in this report.

Methodology:

Scientific studies and papers published concerning IOT security where the main contributor to the requirements reached within this paper.

References:

- Aliyu, F., Sheltami, T., & Shakshuki, E. M. (2018). A detection and prevention technique for man in the middle attack in fog computing. Procedia Computer Science, 141, 24-31
- Alhaidari, F. A., & Alqahtani, E. J. (2020). Securing Communication between Fog Computing and IoT Using Constrained Application Protocol (CoAP): A Survey. J. Commun., 15(1), 14-30

Reviewing Products having capabilities similar to what DTT aims to reach was also a part of the requirement gathering process. Such as Wireshark for its network traffic monitoring capabilities. Namp was also reviewed for its network probing capabilities, and its operating system and hardware characteristics of network devices by observing their network activity.

The last main contributor to this paper, was a brainstorming session with group of young and well educated non experts in the field of security, but holding expertise in other fields, one of the discussion members was a masters degree holder in Astronomical physics, the other was a midlevel web developer.

System Requirements:

DTT Server:

Functional Requirements:

- The system should be able to operate on low powered machines
- The system should be compatible with the Windows OS
- The system should be compatible with Linux OS
- The system should have a GUI
- The system should be able to monitor network activity
- The system should be able to keep a log for network activity
- The system should be able to recognize irregular network activity
- The system should be able to block IPs from network access
- The system should be able to allow IPs network access
- The system should have internet access
- The system should be able to visually notify the user in case of network breach
- The system should be able to send notifications to the mobile application in case of network breach
- The system should allow the user access to the logs
- The system permissions should be conferrable by the user
- The system should take commands through the mobile application
- The system should create a personalized account for the user
- The system should share the user account with the mobile application
- The system should have IP address masking capabilities

- The system should allow the user to activate IP address masking capabilities
- The system should allow the user to deactivate IP masking capabilities
- The system should have data inscription capabilities
- They system should get regular updates to insure everlasting security and compatibility

None-Functional Requirements:

- Client's information should only be stored on the system itself
- All data going out of the home network should be encrypted
- GUI should follow a streamlined and simple design
- Visual design should follows the philosophy of "Form Follows Function"
- Should not feel out of place in a Windows OS machine
- Should not feel out of place in a Linux OS machine
- Notification rate should be set by the user
- The user should be able to disable notifications
- The user should be able to configure DTT to operate as it sees fit with no input from the themselves
- The system should clearly show the user what permissions it has within the home network
- They system should have a force stop option in case of malfunction
- The system should have a simple setup process to not overwhelm the client
- The system should encrypt all data flowing between it and the RCI
- All user data should be kept on the system
- The user should be able to clear all cashed data
- The user should be able to fully reconfigure DTT

- The user should be able to configure multiple presets to fit the circumstance of operation
- The system should not ask the user for any private information(Name, ZIP code, …)
- The system only collects information critical for its operation

DTT RCI:

Functional Requirements:

- The system should have low resource requirements
- The system should compatible with Android devices
- The system should compatible with IOS devices
- The system should follow a modular architecture for easy implementation with future mobile operating systems
- The system should allow for data transfer between it and DTT server
- The system should have data encryption capabilities
- The system should allow the user to pair it with their DTT server
- The system should notify the user when it gets alerts from DTT server
- The system should allow the user to decide what course of action DTT server should take, after alert
- The system should allow the user to configure DTT server remotely
- The system should not hold user information other than needed for operation
- The system should allow the user to access information DTT server have collected

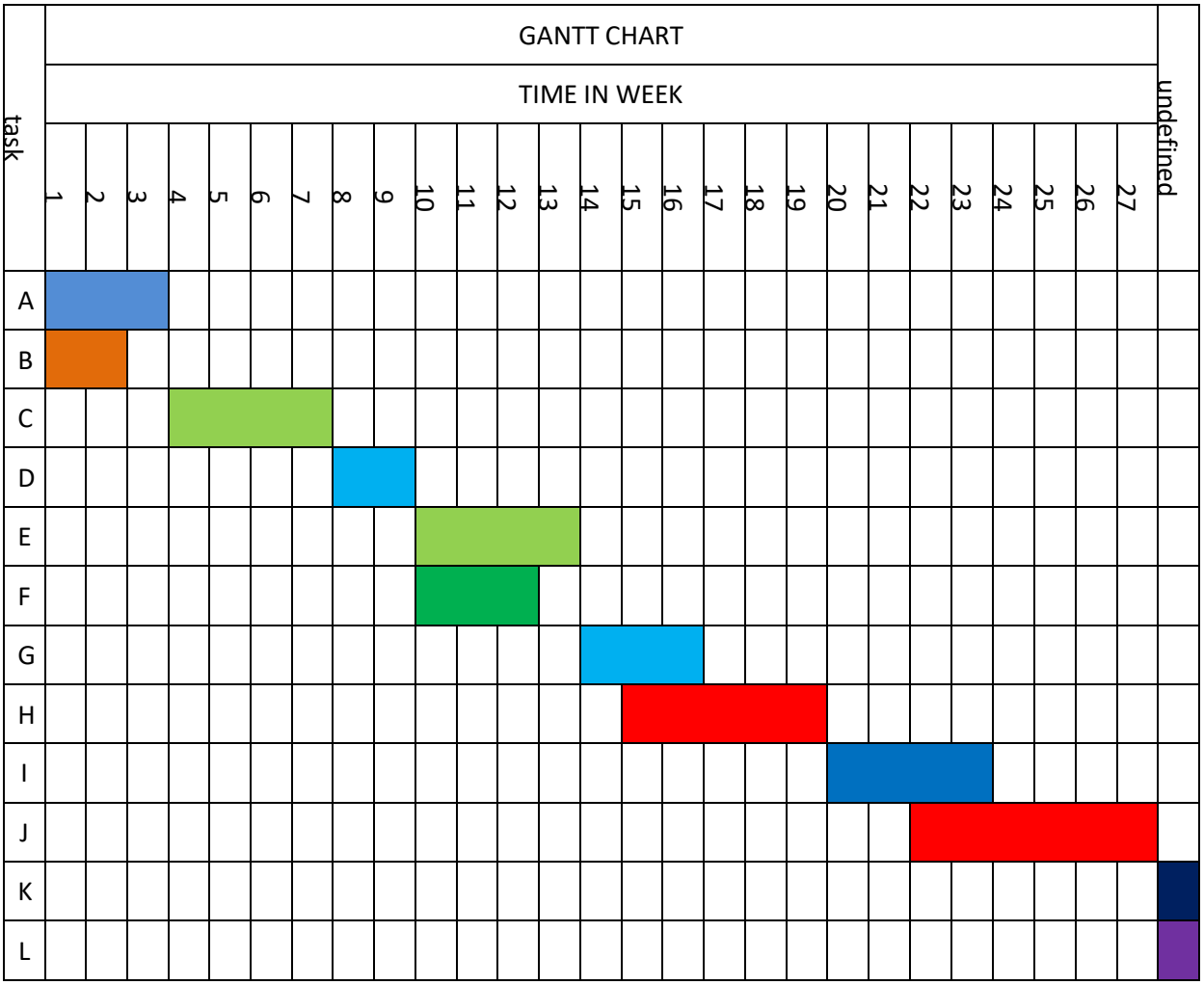
Nonfunctional Requirements:

- Visual design should follow the philosophy of "form follows function"
- The system should visually resemble DTT server The system allow the user to set the rate of notification
- GUI should follow a streamlined and simple design
- The user should be able to disable notifications
- The system should clearly show the user what permissions DTT server has within the home network
- They system should allow the user to use the force stop option of DTT server in case of malfunction
- The system should have a simple pairing process to not overwhelm the client
- The system should encrypt all data flowing between it and DTT sever
- All user data should be kept on the system
- The user should be able to clear all cashed data
- The user should be able to fully reconfigure DTT using the system
- The system should not ask the user for any private information(Name, ZIP code, …)
- The system only collects information critical for its operation

Gantt-Chart:

The following is a Gantt-Chart, specifying the tasks needed to produce our product, and for each task the period (in weeks), it should be finished with in, for simplification of the table, we used letters to signal for tasks and the representation is as follows:

- A: System design
- B: Resource Acquisition
- C: DTT server implementation 1
- D: Testing 1
- E: DTT server implementation 2
- F: DTT RCI implementation
- G: Testing 2
- H: Bug fixing 1
- I: Bata release
- J: Bug fixing 2
- K: Full release
- L: maintenance and updates


GANTT CHART
TIME IN WEEK
task
undefined
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
A
B
C
D
E
F
G
H
I
J
K
L

Conclusion:

After reaching and stating the system requirements, it is quite clear that it will not be an easy task to produce a product such as DTT, but it is not impossible. The design and implementation phases, should keep to the content of this paper, as if these functional and non-functional requirements are respected and implemented in the final product, it will be a well secure, effective, modular, and maintainable product that the market needs.