Secure Software Development Design Document on

Dutch Police Internet Forensics (Government of the Netherlands, n.d.)

Introduction

Day by day, people use the internet in each part of daily life, causing an increase in cybercrimes, affecting public services, businesses, and individual people. In addition, Cybercrimes impacts national security, damage society and cost the government billions every year.

Dutch Police Internet Forensics is responsible for overseeing digital security in the Netherlands. One of its main tasks is to make the Netherlands more resistant to internet crime. (government.nl, 2021).

Project Overview

This project will use specific techniques and python libraries to spoof, capture, and analyse the data and evidence of internet crimes. Out tools can help discover any cyber-crimes and alert any computer intrusion, unauthorised access, child pornography, and any selling of stolen digital data.

The scenario that we will use is:

- ✓ Capturing.
- ✓ Analysing.
- ✓ Alerting.

Technology requirements

We have chosen Python as the coding language to cover the internet forensics tools requirements and to implement it, and we have identified the following requirements:

Hardware

An application server with two network interfaces cards (NIC) connected to a backend database. An access terminal(server/computer) with a host firewall application configured with the necessary access control lists for the required network connectivity.

Software

We will install Linux as an operating system and Python as the programming language on the application server. We will use MySQL as a database server to store all the data capturing from the network adapter, and memory dumps for the data storage. Finally, we will use AWS security tools to configure the access control lists on the network firewall.

The Linux operating system will be used on the client access computer with the required tools to analyse and send alerts about any breach or internet crimes installed on it.

Challenges and Assumptions

We assumed that the Dutch Police Internet Forensics has the authority to install data sensors inside the internet service provider's network without impacting the privacy of unsuspicious data. We also assumed an agreement with internet service providers to share suspicious data with the internet forensic team for further investigation is in place.

Tools and libraries

Python has some great libraries that can be utilised in cybersecurity. The good thing is that most of these libraries are currently being utilised in the cybersecurity area. We are using Python because it is

much simple to learn and user-friendly. The system will be developed in Python on the Codio online IDE platform and PyCharm.

- ✓ Pylint will check for syntax errors and ensure that the code conforms to the PEP 8 style guide.
- ✓ Flask will be used to develop the web application, along with proper authentication.
- ✓ We will use the MySQL database to store the data.
- ✓ Fernet will be used for encrypting and decrypting data.
- ✓ Scrypt/Bcrypt/Argon2 will be used for password hashing.

Documentation

- ✓ Design and architecture - Printed
- ✓ Source code document - Online
- ✓ Readme file - Online
- ✓ Testing documents - Online
- ✓ Help and maintenance – Online

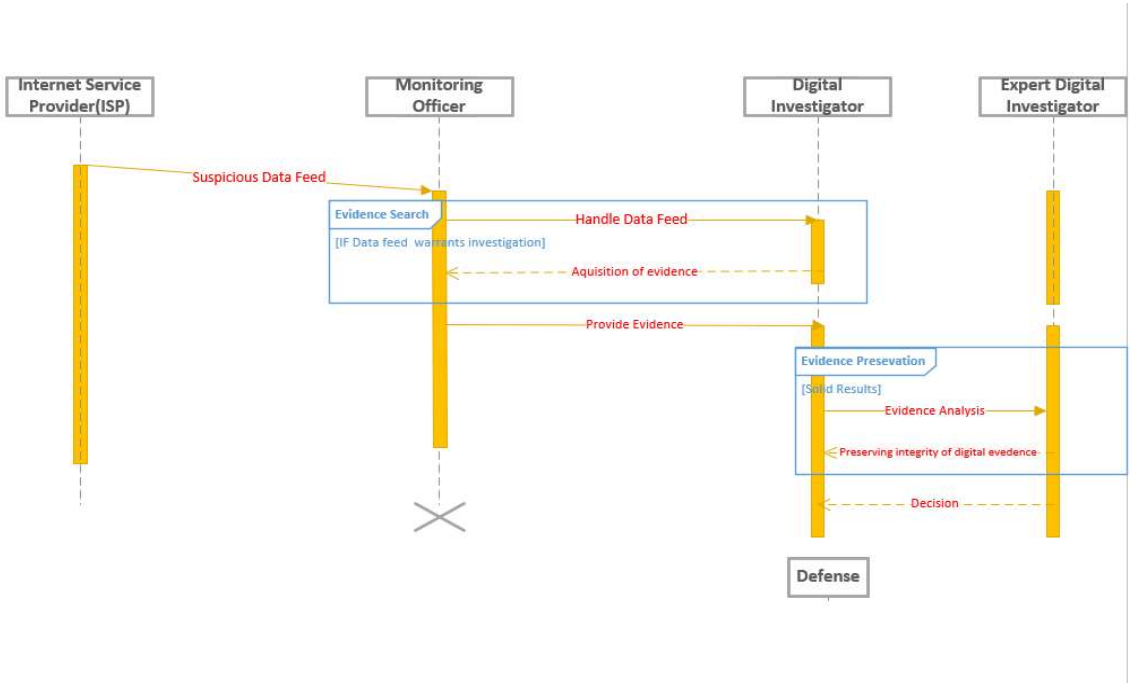
List of patterns and approaches

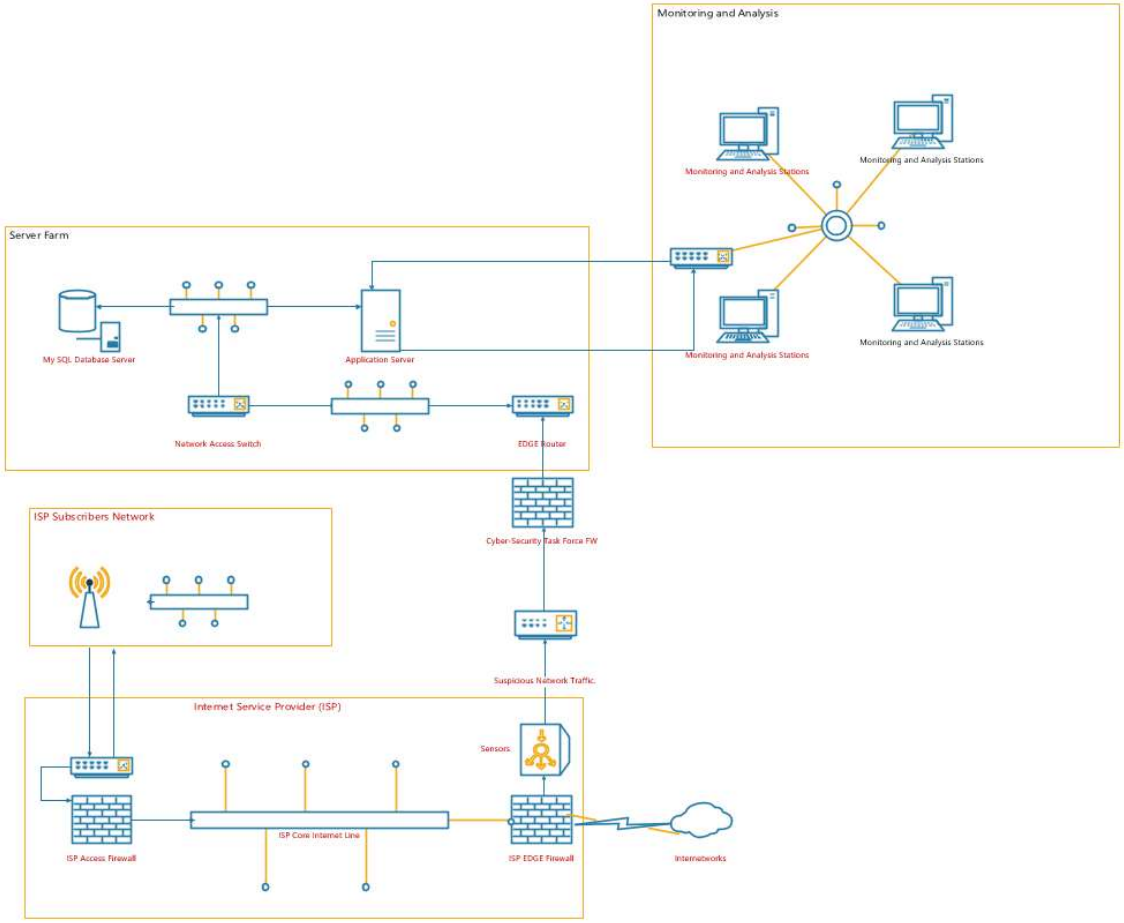
When writing software codes, developers encounter similar problems multiple times within a project. One way to address this is to create design patterns that give engineers a reusable way to solve these problems to achieve the same output structurally for a given project (Anand Butani, 2020)

We intend to use the model-view-controller pattern as it works best with Python coding and our domain, the Dutch police. It has three components, the model, which handles business logic, view for user interface and controller for user input.

Our system will be in line with all GDPR (General Data Protection Regulation) rules. We will also use secure software practices of adhering to least privilege, practice defence-in-depth and keep it simple

System Architecture





References:

- Anand Butani (2020). 5 essential patterns of software architecture. Available from: https://www.redhat.com/architect/5-essential-patterns-software-architecture [Accessed on 29th August 2021]
- Gevernemnt.nl (2021). Fighting cybercrime in the Netherlands. Available from: https://www.government.nl/topics/cybercrime/fighting-cybercrime-in-the-netherlands [Accessed: 28th August 2021]