**OSINT App Test Plan**

Kshiragna Challagundla (ITMS-548), Rahul Nagaraju (ITMS-548), Josh Reginaldo (ITMS-448), Naga Prasath Saravanan (ITMS-548), Brahmantya B. Wardhana (ITMS-448), Sufyan Khan (ITMS-448), Abhishek Deshpande (ITMS-548)

College of Computing, Illinois Institute of Technology

ITMS 448/548: Cyber Security Technologies

Professor Maurice Dawson

December 1, 2023

**Team Members**

| **Members Name** | **Role** |
| --- | --- |
| Sufyan Khan, Rahul Nagaraju, Naga Prasath Saravanan | Project Manager |
| Sufyan Khan | Risk Manager |
| Josh Reginaldo | Developer |
| Brahmantya Wardhana | Developer |
| Abhishek Deshpande | Developer |
| Kshiraguna Challagundla | Data Analyst |

**1.1 Test Risk / Issues**

* **Introduction of Complex Features and/or Implementations**
  + Complex features or implementations can introduce new vulnerabilities or cause errors if not properly programmed in
* **New Developer(s) or Tester(s)** 
  + New developers or testers being introduced to the product might not know how the testing process works,
  + New developer(s) can potentially introduce new vulnerabilities
* **Experienced Developer(s) or Tester(s) Leaving**
  + As these personnel leave, they also bring with them valuable information regarding the inner workings of the product
* **Lack of Documentation**
  + As a product matures and the codebase further grows, documentation of said product should also increase. If documentation is not properly written, new and experienced developers will spend more time figuring out the inner workings of the code and if any errors occur.

**Mitigation Strategies**

* **Introduction of Complex Features and/or Implementations**
  + This can be mitigated by following guidelines set by the company and by being familiar with the documentation, if any
* **Staff Training or Shadowing**
  + To help ease in new personnel, training and/or shadowing will massively help employees be familiar with the product. This will give the new employees an idea of what their responsibilities are and allow them the opportunity to learn any guidelines or procedures from experienced personnel.
* **Proper Documentation**
  + Proper documentation is important to ease the loss of experienced developers or testers as they can include important information about the code and the various tests used to ensure a quality product.

**1.2 Items to be Tested / Not Tested**

| **Test** | **Test Description** | **Date** | **Who** |
| --- | --- | --- | --- |
| **AC-1**  Access Control Policy | Rules for who can access, edit, and finalize changes to the Github repository | **11/13/23** | **Josh Reginaldo** |
| **AC-2**  Account Management | Handle user accounts in the Github and to ensure secure and appropriate management of accounts, reducing risks and maintaining the Github repo integrity | **11/13/23** | **Josh Reginaldo** |
| **AC-5**  Separation of Duties | Identify duties and roles for each member to address the potential for abuse of authorized privileges. | **11/13/23** | **Josh Reginaldo** |
| **AC-6**  Least Privilege | Based on an individual's role, give them the appropriate access to only what they require to their job | **11/13/23** | **Josh Reginaldo** |
| **AC-20**  Use of External Systems | Define guidelines when individuals interact with external systems. This is to safeguard against any unauthorized access or vulnerabilities. | **11/13/23** | **Josh Reginaldo** |
| **AT-2**  Literacy Training and Awareness | Guide non-developers on how to run the application on their own without exposing API keys to the public. | **11/13/23** | **Josh Reginaldo** |
| **AT-3**  Role-Based Training | Provide training based on an individual's role to ensure they are familiar with their responsibilities and tasks | **11/14/23** | **Brahmantya Wardhana** |
| **AT-4**  Training Records | Important documentation that can be used to help guide individuals in their role | **11/14/23** | **Brahmantya Wardhana** |
| **AU-1**  Policy and Procedures | Guidelines for creating and managing audit records. This will help with keeping the project status up-to-date and what has been accomplished. | **11/14/23** | **Brahmantya Wardhana** |
| **AU-2**  Event Logging | Using the Google Suite or Github to see who made changes to what and when those changes were made | **11/14/23** | **Brahmantya Wardhana** |
| **AU-3**  Content of Audit Records | Important information on who was assigned an activity, when it was due, its completion status, and its change history | **11/14/23** | **Brahmantya Wardhana** |
| **MP-3**  Media Marking | Guidelines on how the project items are to be stored, accessed, and shared | **11/15/23** | **Abhishek Deshpande** |
| **RA-3**  Risk Assessment | Conducting a code scan on all related files to analyze potential threats or vulnerabilities | **11/15/23** | **Abhishek Deshpande** |
| **RA-5**  Vulnerability Monitoring and Scanning | Implement use of scanning tools throughout the life cycle of the project to prevent risks or patch vulnerabilities as they show up | **11/15/23** | **Abhishek Deshpande** |
| **SA-3**  System Development Life Cycle | Guidelines or procedures to be followed or executed when a certain stage is reached | **11/15/23** | **Abhishek Deshpande** |
| **SA-10**  Developer Config Management | Base settings that a developer should have to ensure maximum compatibility and security | **11/15/23** | **Abhishek Deshpande** |
| **SA-11**  Developer Testing and Evaluation | Simple test to run on each configuration to ensure that developers have a compatible setup | **11/16/23** | **Josh Reginaldo** |
| **SA-15**  Development Process, Standards, and Tools | Establish development standards when developers are interacting with the code and to allow for maximum compatibility across devices | **11/16/23** | **Josh Reginaldo** |
| **SI-2**  Flaw Remediation | **N/A (** This will not be tested and will be handled alongside RA-5 **)** | **x** | **x** |
| **SI-3**  Malicious Code Protection | **N/A (** This will not be tested and will be handled alongside RA-5 **)** | **x** | **x** |
| **SI-4**  System Monitoring | **N/A (** This will be handled by AC-5/6 as each members will be given the appropriate access to what they require **)** | **x** | **x** |
| **SI-5**  Security Alerts, Advisories, and Directives | **N/A (** Any changes made to the project files have only been assigned to the individuals who were tasked to do them. No outside entities should be able to make changes to the files. **)** | **x** | **x** |
| **SI-7**  Software, Firmware, Info Integrity | **N/A (** This will not be tested and will be handled alongside RA-5, AC 5/6 **)** | **x** | **x** |
| **SI-11**  Error Handling | Print out appropriate error messages for the developer and user to ensure corrective actions are taken | **11/17/23** | **Josh Reginaldo** |
| **SR-2**  Supply Chain Risk Management Plan | **N/A (** This is a small scale project that is not meant to be distributed on a massive scale. The project employs the use of open-source / free APIs from trusted providers **)** | **x** | **x** |
| **SR-8**  Notification Agreements | Choosing a platform to communicate with the the team in the event an incident occurs | **11/17/23** | **Josh Reginaldo** |

**1.3 Test Approach(es)**

We use APIs to get our data and Python as the programming language for our application. With our APIs, we are able to get data on currency exchange rates, country data, crypto coin prices, and sunrise sunset data. The test will be conducted by going through a series of test cases. This series of tests simply involves running the app to check its functionality, testing for proper error handling, and checking for vulnerabilities. Once all test cases have been properly conducted, the testing process is finished.

**1.4 Regulations and Mandates**

* **Quality Standards**
  + Quality Management
  + Risk Management
  + Accepted Sampling
* **Performance Standards**
  + Professionalism
  + Teamwork
  + Communication
  + Time Management
* **Security Standards**
  + Vulnerability Scans
  + Risk Assessment

**1.5 Pass / Fail Criteria**

* **Usability**
  + Interface is user friendly and easy to use
* **Functionality**
* The app functions properly and is able to get desired data

**1.6 Entry / Exit Criteria**

* **Entry**
* The application is complete, meaning it is connected to the 4 apis and is ready for testing
* Test cases are defined and ready

**- Exit**

* The app functions properly without bugs and has proper error handling
* Test cases have been completed
* Any possible vulnerabilities have been identified and fixed if possible

**1.7 Deliverables**

* **Github**
  + Github will serve as the source of the code. This is where we will control the various versions of the app, track who made changes to the code, and manage who has access to the main repository.
* **Test Document**
  + This test document will ultimately be the main deliverable as we will describe the various tests done and their respective outcomes

**1.8 Test Suspension / Resumption Criteria**

* **Suspension Criteria**
  + Tester’s version of the code is out-of-date due to a developer not updating or committing their changes to Github.
  + A Developer is implementing a significant change to the app that is either breaking functionality or has not yet been committed to Github.
* **Resumption Criteria**
  + Tester’s version of the code is aligned with the developers
  + The app is working with all the APIs that have been chosen to be tested.
  + The chosen APIs to be tested are properly sending back data to the program and is being formatted correctly to be displayed in the app

**1.9 Requirements for Testing**

* **Hardware Requirements**
  + Any desktop PC capable of running and compiling Python code
* **Software Requirements**
  + Python Version 3.11.x
  + IDE or Python Compiler
* **Personnel** 
  + Developer(s) who are familiar with the Python programming language
* **Skills**
  + Python
  + Git / Github