Software Requirements Specification Document

Dark web Forensic investigation and mitigation of the data using ML model.

Table of Contents

1. Introduction

1.1 Purpose

The main purpose of this document is to outline the software requirements for **Arjuna**, a dark web forensic investigation tool designed to de-anonymize the uploaders of Information and mitigate the unauthorized distribution of sensitive data on the dark web. This document serves as a guide for developers, stakeholders, and users by specifying the tool's core functionalities, objectives, and legal considerations. It ensures that all parties understand the scope and goals of **Arjuna**, from development through deployment.

1.2 Project Scope

The **Arjuna** tool is designed to address the unauthorized distribution of sensitive data on the dark web by identifying and de-anonymizing the uploaders of such data. The tool will perform the following key activities:

- **Crawl the dark web** to search for relevant data linked to Personally Identifiable Information (PII) and other sensitive content.
- **Scrape and extract data** from dark web sites, focusing on personal, financial, and confidential information.
- **Verify the legitimacy** of the collected data to ensure its relevance and authenticity.
- **Deanonymize IP addresses** associated with data owners to uncover the identities of individuals responsible for uploading unauthorized data.
- **Remove unauthorized or illegal data** from the dark web to mitigate its distribution.
- **Perform final verification** to ensure the sensitive data has been successfully removed from the dark web.
- **Generate a detailed report** summarizing the collected data, the results of deanonymization, and the actions taken to remove and mitigate the risks posed by the unauthorized data.

1.3 Overview

Arjuna is a tool designed to combat dark web threats. It identifies and de-anonymizes those who upload sensitive data, removes harmful content, and provides insights for law enforcement. Key features include secure tunneling, dark web crawling, data analysis, IP de-anonymization, data removal, and report generation.

2. General Description

The objective of this project is to develop a comprehensive solution for forensic investigation and mitigation of unauthorized data distribution on the dark web, utilizing advanced machine learning algorithms. **Arjuna** encompasses the following activities:

- **Tunneling**: Establish a secure tunnel to access the dark web, ensuring all communications are encrypted and anonymous.
- **Crawling**: Crawl the dark web to search for relevant URL data, navigating through hidden sites and directories.
- **Scraping**: Extract sensitive information from the crawled data, such as personal identifiers, financial records, and confidential information.
- **Data Classification**: Classify the scraped data into predefined parameters. This classified data is then fed into a machine learning model for further processing.
- **IP De-anonymization**: Trace the IP addresses associated with the unauthorized data and attempt to de-anonymize the owner who is responsible for uploading it.
- **Data Removal**: Identify unauthorized or illegal data and take appropriate steps to remove it from the dark web.
- **Verification**: After removal, verify that the sensitive data is no longer accessible on the dark web.
- **Report Generation**: Generate a detailed report summarizing the collected data, deanonymization results, and actions taken to remove and mitigate the risks associated with the unauthorized data.

Project Phases:

1. **Data Acquisition:**
    - Implement efficient crawling and scraping techniques.
    - Develop automated bots for repetitive tasks.
2. **Data Analysis:**
    - Classify scraped data using machine learning models.
    - Employ advanced techniques for de-anonymization.
3. **Mitigation and Verification:**
    - Remove unauthorized data from the dark web.
    - Verify the effectiveness of removal efforts.

3. Functional Requirements

1. Dark Web Crawling

- **Efficient crawling:** Arjuna should be capable of efficiently crawling various dark web markets and forums, including those that use dynamic content and obfuscation techniques.
- **Customizable parameters:** The tool should allow users to specify crawling parameters such as keywords, categories, and timeframes.
- **Handling CAPTCHAs and other challenges:** Arjuna should be able to handle CAPTCHAs and other common challenges encountered during web crawling.
- **Integration with dark web directories:** The tool should be able to integrate with popular dark web directories and search engines.

2. Data Extraction

- **Extraction of relevant data:** Arjuna should be able to extract a wide range of relevant data from crawled content, including text, images, metadata, and links.
- **Support for various file formats:** The tool should support various file formats, such as PDF, DOCX, and HTML.
- **Extraction of hidden data:** Arjuna should be able to extract hidden data, such as metadata embedded within images or documents.
- **Data normalization:** The tool should normalize extracted data to ensure consistency and facilitate analysis.

3. Data Analysis

- **Machine learning algorithms:** Arjuna should leverage advanced machine learning algorithms to analyze extracted data and identify patterns, anomalies, and potential threats.
- **Natural language processing:** The tool should employ natural language processing techniques to understand and analyze textual data.
- **Data visualization:** Arjuna should provide intuitive data visualization tools to help users understand analysis results.
- **Integration with external databases:** The tool should be able to integrate with external databases and services for additional data analysis and context.

4. IP De-anonymization

- **IP address tracing:** Arjuna should be able to trace IP addresses associated with uploaded content.
- **Integration with IP intelligence databases:** The tool should integrate with IP intelligence databases to obtain additional information about IP addresses.
- **Anonymization techniques:** Arjuna should be able to identify and counter common anonymization techniques used on the dark web.
- **De-anonymization methods:** The tool should employ various de-anonymization methods, such as geolocation, historical data analysis, and social network analysis.

5. Data Removal

- **Identification of unauthorized content:** Arjuna should be able to identify unauthorized or illegal content based on predefined criteria or user-defined rules.
- **Removal requests:** The tool should allow users to submit removal requests for identified content.
- **Automated removal:** Arjuna should be capable of automatically removing content from the dark web, if possible.
- **Verification of removal:** The tool should verify that the requested content has been successfully removed.

6. Report Generation

- **Comprehensive reports:** Arjuna should generate detailed reports summarizing the results of the investigation, including extracted data, analysis findings, and actions taken.
- **Customizable templates:** The tool should allow users to customize report templates to meet specific requirements.
- **Export options:** Reports should be exportable in various formats, such as PDF, CSV, and HTML.
- **Integration with law enforcement systems:** Reports should be easily exportable to law enforcement systems for further investigation.

4. Non-Functional Requirements

1. Storage

- **Capacity:** Arjuna should have sufficient storage capacity to accommodate the vast amount of data scraped from the dark web, including text, images, and metadata.
- **Durability:** The storage solution should be reliable and resilient to data loss or corruption.
- **Accessibility:** Data should be easily accessible for analysis and processing.
- **Scalability:** The storage solution should be scalable to accommodate future growth and increased data volumes.

2. EC2 Instance

- **Compute power:** Arjuna should be deployed on an AWS EC2 instance with sufficient compute power to handle the computational demands of data analysis, machine learning, and real-time processing.
- **Memory:** The instance should have adequate memory to store data, models, and intermediate results.
- **Storage:** The instance should be configured with appropriate storage options to meet the tool's data storage needs.

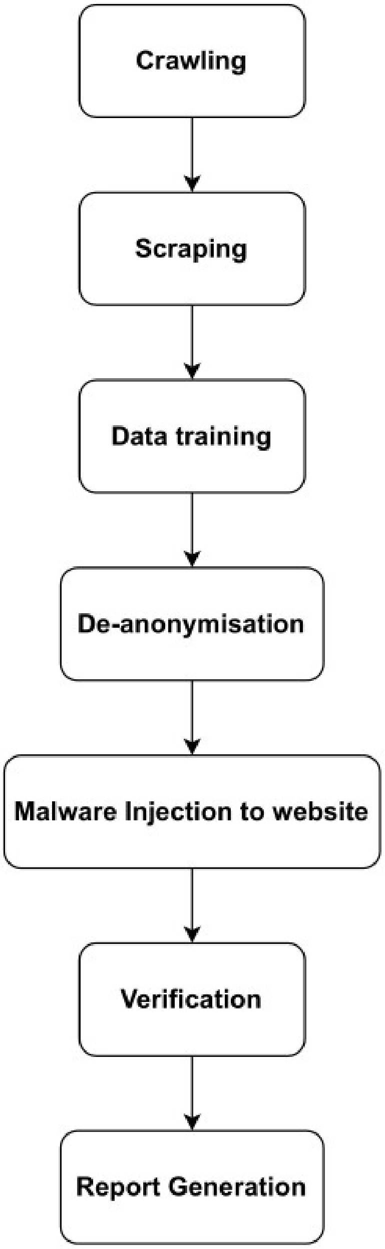
3. GPU Machine

- **Processing power:** Arjuna should leverage a high-performance GPU to accelerate machine learning training and inference, especially for tasks involving deep learning models.
- **Compatibility:** The GPU should be compatible with popular machine learning frameworks and libraries.
- **Cost-effectiveness:** The GPU should provide a good balance of performance and cost.

4. Cloud Infrastructure

- **AWS platform:** Arjuna should be deployed on the AWS cloud platform to leverage its scalability, reliability, and comprehensive suite of services.
- **Network configuration:** The cloud infrastructure should be configured to allow secure access to the dark web and external resources.
- **Cost optimization:** Strategies should be implemented to optimize resource usage and minimize costs.

5. Project Flow



6. Preliminary Schedule

Phase 1: Data Acquisition

- Task 1: Tunneling Infrastructure Setup
- Task 2: Crawler Development
- Task 3: Scraper Development

Phase 2: Data Analysis

- Task 4: Classification Model Development
- Task 5: De-anonymization Techniques Research
- Task 6: De-anonymization Tool Development

Phase 3: Threat Mitigation

- Task 7: Removal Algorithm Development
- Task 8: Verification Process Implementation

Overall Project Duration: 3 Months (1st Oct 2024 - 1st Jan 2024)

7. Appendices

References

- **CRATOR: Dark Web Crawler:** [Link] - A tool for crawling and extracting data from the dark web.
- **Cloudflare: What is Tunneling?:** [Link] - Secure tunneling methods for data transmission.
- **ACHE: Adaptive Crawler:** [Link] - Java-based crawler for hidden web content.
- **TorBot: Python-based Dark Web Crawler:** [Link] - Tool for collecting and analyzing dark web data.

8. Other Requirements

- **Data Model:** Design a robust data model to efficiently store and manage extracted data, including crawled websites, extracted information, and analysis results.
- **Indexes:** Create appropriate indexes to optimize query performance and improve data retrieval efficiency.
- **Data Security:** Implement encryption and access controls to protect sensitive data.

Legal and Ethical Considerations

- **Compliance:** Ensure compliance with relevant laws and regulations, such as data privacy laws (GDPR, CCPA) and intellectual property laws.
- **Ethical Use:** Adhere to ethical principles and avoid any harmful or malicious activities.

Reuse Objectives

- **Modularity:** Design Arjuna with a modular architecture to facilitate component reuse in other projects.
- **API Design:** Provide well-defined APIs for integration with other systems.
- **Documentation:** Create comprehensive documentation to aid in understanding and reusing Arjuna's components.

Additional Considerations

- **Scalability:** Design Arjuna to handle increasing workloads and data volumes efficiently.
- **Performance Optimization:** Implement techniques to optimize performance, such as caching, parallelization, and distributed computing.
- **Integration with Third-Party Tools:** Consider integrating with other tools or services for additional functionalities (e.g., visualization tools, threat intelligence platforms).
- **User Experience:** Focus on creating a user-friendly interface with intuitive navigation and helpful visualizations.

Appendix A: Glossary

Dark Web: A portion of the internet that is not indexed by search engines and requires specific software to access.

De-anonymization: The process of identifying an individual or entity associated with an anonymous online identity.

Machine Learning: A type of artificial intelligence that allows computers to learn from data and improve their performance over time.

Data Scraping: The process of extracting data from websites.

IP Address: A unique numerical label assigned to each device connected to the internet.

Malware: Malicious software designed to harm computer systems or steal data.

Tunneling: A technique used to create a secure connection between two networks over an insecure network.

Crawling: The process of systematically exploring the web to discover and index content.

Scraping: The process of extracting data from websites.

Data Classification: The process of categorizing data into predefined categories or labels.

De-anonymization: The process of identifying an individual or entity associated with an anonymous online identity.

Data Removal: The process of removing unauthorized or illegal content from the dark web.

Verification: The process of confirming that the desired action has been successfully completed.

Report Generation: The process of creating a detailed report summarizing the findings of an investigation.

Appendix B: Issues List

1. Data Privacy and Ethical Considerations:

- How to ensure compliance with data privacy regulations (e.g., GDPR, CCPA)?
- How to address ethical concerns related to data collection and analysis?

2. Scalability and Performance:

- How to handle large-scale data sets and ensure efficient performance?
- What strategies can be used to optimize resource utilization and minimize costs?

3. Machine Learning Model Selection:

- Which machine learning algorithms are best suited for the specific tasks involved in Arjuna?
- How to evaluate and select the optimal model for data classification and de-anonymization?

4. Data Quality and Validation:

- How to ensure the quality and accuracy of the scraped data?
- What techniques can be used to clean and preprocess the data?

5. Integration with External Systems:

- How to integrate Arjuna with law enforcement databases and other relevant systems?
- What APIs and protocols should be used for data exchange?

6. User Interface and Experience:

- How to design a user-friendly and intuitive interface?
- What visualization tools should be incorporated to help users understand the results?

7. Security and Privacy:

- How to protect sensitive data from unauthorized access and breaches?
- What security measures should be implemented to prevent malware and other threats?

8. Legal and Regulatory Compliance:

- How to ensure compliance with relevant laws and regulations, such as copyright and intellectual property laws?

9. Deployment and Maintenance:

- What deployment strategies should be used for Arjuna?
- How to ensure the tool's ongoing maintenance and updates?

10. Resource Allocation:

- How to allocate resources effectively, including hardware, software, and personnel?