Secure Software Design & Engineering (CY-321)

Threat Modeling & Risk Assessment

Open-Source OSINT Tool for Investigators (`AUSAINT`)

Group Members:
Afnan Bin Abbas (2022048)
Muhammad Umar Maqsood (2022447)
Shamina Durrani (2022543)
Aqib Shakeel (2022104)

March 14, 2025

Ghulam Ishaq Khan Institute of Engineering Sciences and Technology

1 Attack Vectors & Risk Levels

To ensure the security and reliability of the OSINT Investigative Suite, we employ the **STRIDE** threat modeling framework. STRIDE helps identify key risks such as **Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege**. By applying this framework, we systematically analyze potential attack vectors, assess their impact, and develop targeted mitigation strategies to ensure a secure and reliable OSINT tool.

Attack Vector	**Description**	**Risk Level**
SQL Injection & XSS	Attackers may inject malicious SQL queries or cross-site scripting to access sensitive data.	High
API Key Exposure	If API keys are leaked, attackers could abuse premium OSINT services.	High
Data Integrity Violation	Unauthorized modification of intelligence data by an attacker.	High
Brute-Force Authentication Attacks	Attackers may attempt to gain access using credential stuffing or dictionary attacks.	Medium
Information Disclosure	Sensitive data may be leaked if not encrypted properly.	High
Denial of Service (DoS)	Attackers may overload the system with excessive OSINT queries, causing service disruption.	Medium
Third-Party Service Compromise	The OSINT tool relies on external APIs, which could be compromised.	Medium
Phishing & Social Engineering	Attackers may manipulate users into revealing login credentials or API keys.	Low
Insufficient Logging & Monitoring	Lack of proper logs may allow attackers to bypass detection.	Medium

Table 1: Attack Vectors and Risk Assessment

2 Security Mitigation Strategies

To address the identified threats and ensure the robustness of the OSINT Investigative Suite, we propose the following mitigation strategies. These measures are designed to secure user data, prevent abuse, and maintain the integrity of intelligence gathered by the tool.

Attack Vector	**Mitigation Strategy**
SQL Injection & XSS	Implement strict input validation and use ORM to prevent SQL injection. Sanitize all inputs for XSS.
API Key Exposure	Store API keys securely using environment variables. Implement role-based access control.
Data Integrity Violation	Use cryptographic hashing (SHA-256) to ensure data integrity.
Brute-Force Authentication Attacks	Implement account lockout and CAPTCHA after multiple failed login attempts.
Information Disclosure	Encrypt stored intelligence data and enforce HTTPS for secure data transmission.
Denial of Service (DoS)	Implement rate limiting and request throttling to mitigate excessive API requests.
Third-Party Service Compromise	Regularly audit third-party APIs and implement fallback mechanisms.
Phishing & Social Engineering	Enforce 2FA for authentication and educate users on phishing awareness.
Insufficient Logging & Monitoring	Enable real-time logging, monitor logs for anomalies, and set up alert notifications.

Table 2: Security Mitigation Strategies