Business Requirements Document (BRD)

Automated Threat Modeling System (ATM)

1. Introduction

1.1 Background

As software ecosystems evolve, they often mirror the organic intricacies and symbioses found in biological systems. Modern applications, while enhancing interactivity and functionality, are now comprised of multiple microservices, integrated third-party solutions, cloud infrastructures, and a plethora of devices. While these architectures provide scalability and flexibility, they also open up a multitude of potential weak points. Traditional threat modeling methodologies, confined by manual limitations, no longer suffice. The onset of the ATMS is our answer to these emerging challenges, providing a dynamic, adaptive, and timely response to contemporary cyber threats.

1.2 Objective

To meticulously design and deploy the ATMS—a pioneering tool aiming to redefine threat modeling. Through this, organizations will not only identify and evaluate threats but also pre-emptively strategize, focusing on the most probable and impactful vulnerabilities.

1.3 Scope

* The ATMS is not merely a tool; it's a paradigm shift. It is envisioned to:
* Centralize and standardize threat modeling practices.
* Foster a proactive security culture, moving from reactive patches to predictive modeling.
* Weave into the fabric of CI/CD pipelines, ensuring that security is a continuous process.
* Remain agile, adjusting to new threat vectors, and accommodating the ever-evolving compliance landscapes.

2. Business Requirements

2.1 Business Objective

The ATMS envisions:

* Modernizing the threat modeling paradigm.
* Reducing the duration between threat identification and resolution.
* Embedding a shift-left security approach within teams.
* Seamless integration within prevalent software infrastructures.

3. Stakeholders

Development Teams: Directly interfacing with the tool during the software development lifecycle.

Security Analysts: Overseeing threat modeling, analysis, and mitigation strategies.

IT Operations: Ensuring seamless integration within existing infrastructures.

Compliance Teams: Ensuring outputs adhere to regulatory requirements.

Senior Management: Evaluating the tool's ROI and strategic alignment with organizational goals.

4. System Overview

The ATMS will provide a centralized dashboard, integrating various modules like threat discovery, assessment, reporting, and mitigation. While being cloud-agnostic, it will also offer integrations with popular CI/CD platforms, threat intelligence feeds, and compliance checklists.

5. Data Flow

Ingesting data from source code repositories, architectural diagrams, and third-party integrations, ATMS will process this through its rule-based engines. Outputs, in the form of threat landscapes, reports, and recommendations, will then be channeled to the appropriate stakeholders.

6. Assumptions & Constraints

Assumptions:

- Stakeholders have a basic understanding of threat modeling.

- The software ecosystem remains compatible with the tool.

Constraints:

- The tool will not decipher proprietary architectural diagram formats.

- Real-time processing might experience minimal lags during peak loads.

7. Risks & Mitigation

Risks:

- Incomplete threat modeling due to evolving threat vectors.

- Potential resistance from development teams due to workflow changes.

Mitigation:

- Regular updates and synchronization with threat intelligence.

- Comprehensive training and integration assistance.

8. Acceptance Criteria

Functional Requirements

FR1 Threat Discovery:

Successful identification and cataloging of over 95% of known threat vectors in test environments.

FR2 Application Compatibility:

Demonstrable support and flawless functionality across a minimum of three major application architectures (e.g., web, mobile, IoT).

FR3 Diagram Parsing:

Accurate interpretation and data extraction from a minimum of five industry-standard architectural diagram formats.

FR4 Threat Categorization:

Alignment and categorization accuracy of at least 98% when benchmarked against standards like STRIDE and OWASP.

FR5 Threat Assessment:

Accurate risk scoring with a margin of error less than 5% when compared against expert manual assessments.

FR6 Reporting:

Generation of segmented reports with clear visual representations and actionable insights, requiring less than 3 clicks for navigation.

FR7 Integration:

Seamless connection with a minimum of three external threat intelligence sources without data discrepancies.

FR8 Mitigation Proposals:

At least 90% of the countermeasure suggestions provided should be ratified as effective by security experts during tests.

FR9 CI/CD Embedment:

Successful embedding within three major CI/CD platforms with real-time threat modeling demonstrations.

FR10 Regulatory Alignment:

Clear adherence and compliance verification against at least two major global standards.

FR11 User Management:

Effective implementation of Role-Based Access Controls (RBAC) with zero breaches in mock penetration tests.

Non-functional Requirements

NFR1 Design:

Scalable architecture capable of accommodating a 50% increase in load without degradation in performance.

NFR2 Security:

Withstand a minimum of three industry-standard penetration tests with zero critical vulnerabilities.

NFR3 Reliability:

Demonstrate a consistent uptime of 99.99% over a monitored period of 6 months.

NFR4 UI/UX:

Positive feedback from at least 85% of users regarding usability and experience in pilot tests.

NFR5 Performance:

Exhibit response times below two seconds for primary functions under normal load.

NFR6 Scalability:

Show zero degradation in performance when scaled up to handle an increased user load of 40% during stress tests.

9. System Constraints

9.1 Hardware Constraints

Resource Consumption: While ATMS is designed for optimal performance, there may be constraints on extremely resource-restricted environments.

Network: Continuous, high-speed internet connectivity is essential for real-time threat modeling and synchronization with threat intelligence feeds.

9.2 Software Constraints

Compatibility: The ATMS is designed to be platform-agnostic but may face compatibility issues with very outdated operating systems or software.

Third-party Integrations: ATMS relies on API integrations with other platforms. Changes in these APIs might temporarily disrupt functionalities until updates are made.

9.3 Data Constraints

Data Formats: ATMS can process standard architectural diagram formats and common code repositories, but proprietary formats may not be supported.

Data Storage: There's a constraint on the retention period of historical threat modeling data, beyond which it will be archived.

---

10. User Constraints

10.1 Training

Users must undergo basic training to fully leverage ATMS's capabilities, as improper usage could lead to inaccurate threat modeling.

10.2 Access Restrictions

For security reasons, not all functionalities will be available to all users. Access is granted based on roles and responsibilities.

10.3 Feedback Loop

Users are encouraged to provide feedback but might experience a delay in response during high submission periods.

---

11. Conclusion

The ATMS, as conceptualized, represents a comprehensive step forward in the domain of threat modeling. Its automated, agile, and proactive approach positions it as a much-needed solution in today's dynamic cyber threat landscape. While there are inherent constraints and limitations, as with any system, the overarching goal is to seamlessly blend into software development lifecycles, enhancing security postures and fostering a proactive security culture. As we move forward with its implementation, the emphasis will be on continuous evolution, ensuring it remains at the forefront of threat modeling technologies.