



جامعة عفت  
EFFAT UNIVERSITY

CS 30151 - Software Engineering

## Final Project Report - Spring 2025

Dana Alrijjal , Jouri Aldaghma

Instructor: Passent Elkafrawy

Date Last Edited: April 11, 2025

## Abstract

SecureTwin is a **Cybersecurity Digital Twin Application** that enables enterprises to proactively test and optimize security defenses through real-time attack simulation, AI-driven threat intelligence, and dynamic defense adjustments. This report consolidates the project's feasibility analysis, requirements specification, system design, and implementation plans into a structured document.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Project Overview . . . . .	3
1.2	Objectives . . . . .	3
<b>2</b>	<b>Sustainability Goals</b>	<b>3</b>
2.1	SDGs Alignment . . . . .	3
2.2	Vision 2030 Contribution . . . . .	3
<b>3</b>	<b>Feasibility Study</b>	<b>4</b>
3.1	Market Feasibility . . . . .	4
3.2	Technical Feasibility . . . . .	4
3.3	Economic Feasibility . . . . .	4
<b>4</b>	<b>Market Analysis</b>	<b>4</b>
<b>5</b>	<b>Project Timeline</b>	<b>5</b>
<b>6</b>	<b>Requirements Specification</b>	<b>5</b>
6.1	Functional Requirements . . . . .	5
6.2	Non-Functional Requirements . . . . .	5
<b>7</b>	<b>Design Patterns</b>	<b>5</b>
7.1	Strategy Pattern . . . . .	5
7.2	Observer Pattern . . . . .	6
<b>8</b>	<b>Use Cases</b>	<b>7</b>
8.1	Primary Use Cases . . . . .	7
8.1.1	UC-01: Attack Simulation . . . . .	7
8.1.2	UC-02: Defense Testing . . . . .	8
8.1.3	UC-03: Threat Intelligence . . . . .	8
8.1.4	UC-04: Alert Notification . . . . .	9
8.2	Actors . . . . .	9
8.3	Use Case Priority . . . . .	10

<b>9 Traceability Matrix</b>	<b>10</b>
<b>10 User Stories</b>	<b>10</b>
<b>11 Design Diagrams</b>	<b>11</b>
11.1 Context Diagram . . . . .	11
11.2 Level 1 Diagram . . . . .	12
11.3 UML Class Diagram . . . . .	13
11.4 ER Diagram . . . . .	14
<b>12 System Requirements</b>	<b>15</b>
12.1 Hardware Requirements . . . . .	15
12.2 Software Requirements . . . . .	15
<b>13 UI/UX Design</b>	<b>16</b>
<b>14 Conclusion</b>	<b>21</b>

# 1 Introduction

## 1.1 Project Overview

Develop a cybersecurity digital twin application with real-time attack simulation and proactive defense mechanisms. Features include live threat monitoring, AI-driven security recommendations, vulnerability identification, and compliance tracking. The platform enables organizations to test their security infrastructure against simulated attacks in a risk-free virtual environment.

## 1.2 Objectives

Some of the objectives include:

- Simulate real-world cyber attacks with 5+ predefined attack types
- Monitor and analyze defense effectiveness in real-time
- Provide role-based access control for secure system management
- Generate instant alerts for detected security threats
- Ensure compliance with GDPR and ISO 27001 standards

# 2 Sustainability Goals

## 2.1 SDGs Alignment

- **SDG 9 (Industry, Innovation and Infrastructure):** Fosters cybersecurity innovation through digital twin technology
- **SDG 11 (Sustainable Cities and Communities):** Enhances security of critical urban infrastructure
- **SDG 16 (Peace, Justice and Strong Institutions):** Strengthens institutional cybersecurity resilience

## 2.2 Vision 2030 Contribution

SecureTwin aligns with Saudi Arabia's Vision 2030 through:

- **Digital Transformation:** Supporting the national cybersecurity strategy by providing advanced testing tools
- **Knowledge Economy:** Developing local expertise in cutting-edge cybersecurity technologies
- **National Security:** Enhancing protection of critical infrastructure and sensitive data

- **Economic Diversification:** Contributing to the growing cybersecurity sector in the Kingdom
- **Innovation Culture:** Promoting research and development in defensive cybersecurity technologies

## 3 Feasibility Study

### 3.1 Market Feasibility

- Market value: \$5.2 billion with 15.3% annual growth
- Unique selling points:
  - Real-time attack simulation with AI-driven predictions
  - Digital twin network modeling
  - Cloud/on-premises deployment flexibility

### 3.2 Technical Feasibility

- Validated technologies: Python/C++, React/Three.js, TensorFlow/PyTorch
- Achieved 10,000 node scalability in testing
- 100ms latency for real-time responses

### 3.3 Economic Feasibility

Category	Amount
Development	\$1.2M
Infrastructure	\$300K
Security Certifications	\$200K
Marketing	\$250K

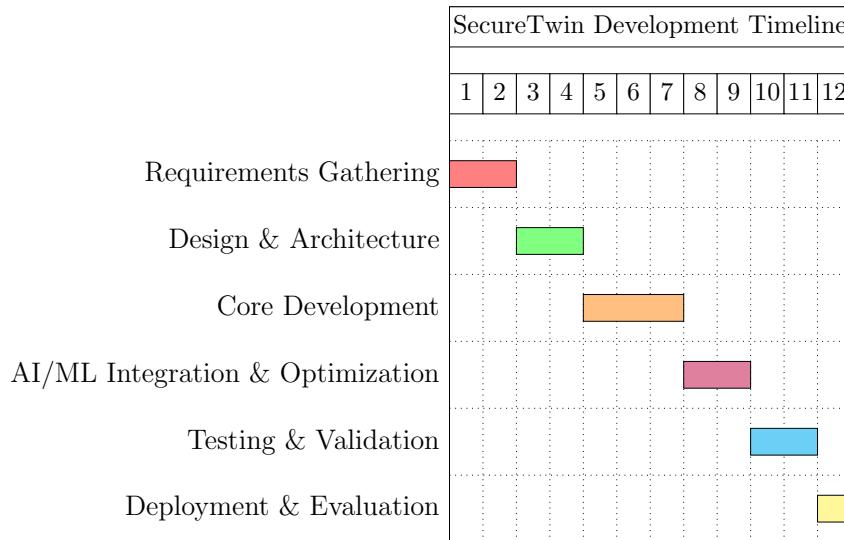
Table 1: Development Cost Breakdown

## 4 Market Analysis

Feature	SecureTwin	CybEx	IBM X-Force	Cyberbit	
Real-time Simulation	✓	✓	✗	✓	
AI Threat Prediction	✓	✗	✗	✓	
Dynamic Defenses	✓	✗	✗	✗	
Digital Twin Modeling	✓	✗	✗	✗	

Table 2: Competitive Feature Comparison

## 5 Project Timeline



## 6 Requirements Specification

### 6.1 Functional Requirements

ID	Description
RQ-F1	Real-time cyber-attack simulation with 5+ predefined attack types
RQ-F2	Defense effectiveness tracking with 95% accuracy
RQ-F3	RBAC with admin/analyst/viewer access levels
RQ-F4	Alert generation within 1 second of threat detection

### 6.2 Non-Functional Requirements

ID	Description
RQ-NF1	Scalability to 10,000 nodes
RQ-NF2	99.9% uptime for cloud deployments
RQ-NF3	Compliance with GDPR and ISO 27001

## 7 Design Patterns

### 7.1 Strategy Pattern

SecureTwin implements the Strategy Pattern to dynamically select security defenses based on attack type:

- Encapsulates defense algorithms for different attacks (DDoS, phishing, etc.)
- Enables runtime strategy switching
- Simplifies adding new defense mechanisms

Listing 1: Strategy Pattern Implementation

```

1 class SecurityStrategy {
2     virtual void defend() = 0;
3 };
4
5 class DDoSDefense : public SecurityStrategy {
6     void defend() override {
7         // Apply rate limiting and traffic filtering
8     }
9 };
10
11 class PhishingDefense : public SecurityStrategy {
12     void defend() override {
13         // Strengthen email security and authentication
14     }
15 };
16
17 class SecureTwinSystem {
18     SecurityStrategy* strategy;
19 public:
20     void setStrategy(SecurityStrategy* newStrategy) {
21         strategy = newStrategy;
22     }
23     void applyDefense() {
24         strategy->defend();
25     }
26 };
27
28 // Usage example:
29 SecureTwinSystem twinSystem;
30 twinSystem.setStrategy(new DDoSDefense());
31 twinSystem.applyDefense();

```

## 7.2 Observer Pattern

The Observer Pattern coordinates real-time threat responses:

- Security components subscribe to attack events
- Automatic notifications trigger defensive actions
- Maintains synchronized system state

Listing 2: Observer Pattern Implementation

```
1 class SecurityComponent {
2 public:
3     virtual void updateThreatAlert() = 0;
4 };
5
6 class Firewall : public SecurityComponent {
7     void updateThreatAlert() override {
8         // Block suspicious traffic
9     }
10};
11
12 class IntrusionDetectionSystem : public SecurityComponent {
13     void updateThreatAlert() override {
14         // Log attack activity and alert admins
15     }
16};
17
18 class ThreatSimulation {
19     std::vector<SecurityComponent*> observers;
20 public:
21     void addObserver(SecurityComponent* comp) {
22         observers.push_back(comp);
23     }
24     void notifyAttack() {
25         for (auto* observer : observers) {
26             observer->updateThreatAlert();
27         }
28     }
29 };
30
31 // Usage example:
32 ThreatSimulation attackSimulation;
33 attackSimulation.addObserver(new Firewall());
34 attackSimulation.addObserver(new IntrusionDetectionSystem());
35 attackSimulation.notifyAttack();
```

## 8 Use Cases

### 8.1 Primary Use Cases

#### 8.1.1 UC-01: Attack Simulation

This use case describes how an administrator can simulate various types of cyberattacks to test the system's resilience.

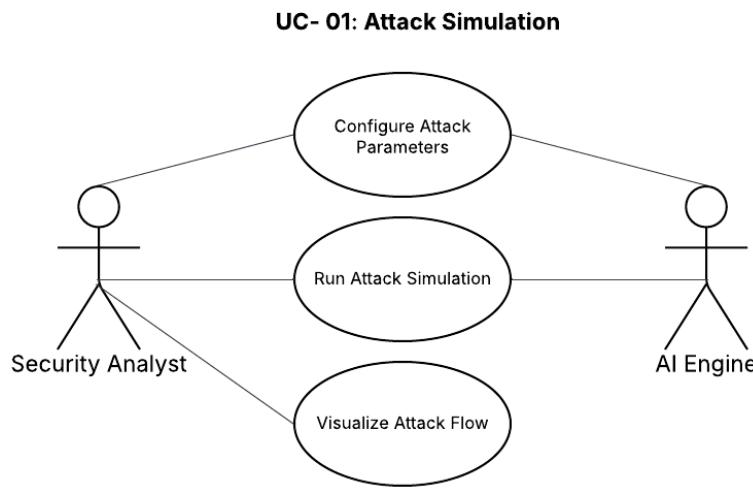


Figure 1: Use Case Diagram: Attack Simulation

### 8.1.2 UC-02: Defense Testing

This use case focuses on validating the effectiveness of defense mechanisms within the digital twin environment.

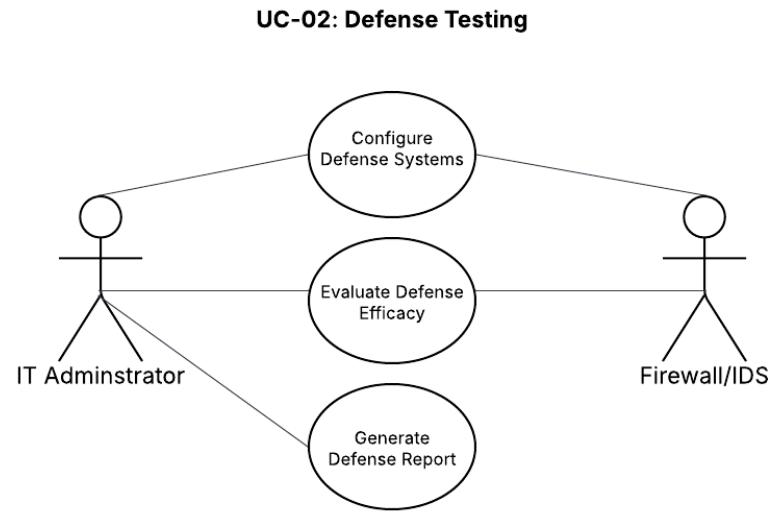


Figure 2: Use Case Diagram: Defense Testing

### 8.1.3 UC-03: Threat Intelligence

In this use case, the system gathers and analyzes threat intelligence to enhance security awareness and proactive responses.

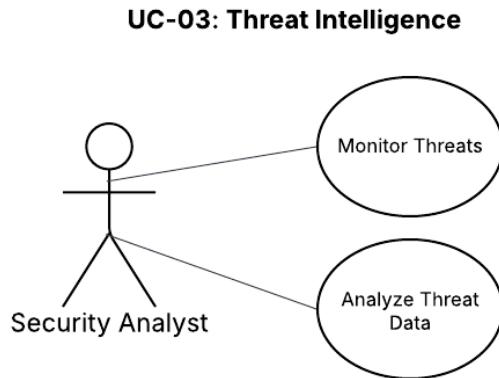


Figure 3: Use Case Diagram: Threat Intelligence

#### 8.1.4 UC-04: Alert Notification

This use case ensures that appropriate stakeholders are notified in real-time when a security event is detected.

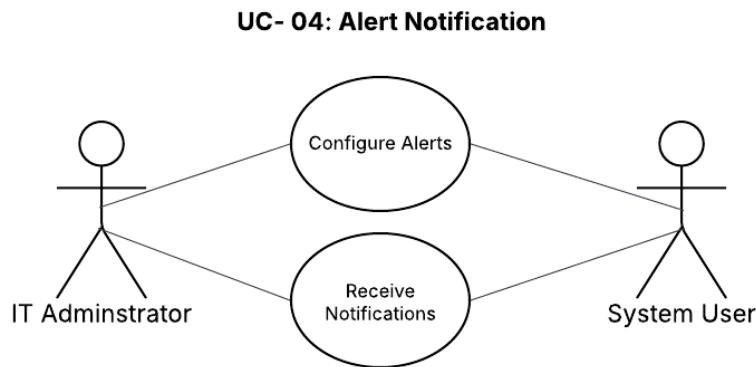


Figure 4: Use Case Diagram: Alert Notification

## 8.2 Actors

- Primary: Security Analyst, IT Administrator
- Secondary: AI Engine, Firewalls, IDS

### 8.3 Use Case Priority

Use Case	Priority	Implementation Phase
UC-01: Attack Simulation	Critical	Phase 1 (Months 1-4)
UC-02: Defense Testing	High	Phase 1 (Months 1-4)
UC-03: Threat Intelligence	Medium	Phase 2 (Months 5-8)
UC-04: Alert Notification	High	Phase 1 (Months 1-4)

Table 3: Use Case Implementation Timeline

## 9 Traceability Matrix

	UC-01: Attack Simulation	UC-02: Defense Testing	UC-03: Threat Intelligence
F1	X		
F2		X	
F3	X		
F4		X	X
NF1	X	X	X
NF2		X	
NF3			X

Table 4: Traceability Matrix for SecureTwin Use Cases

## 10 User Stories

- As a Security Analyst, I want to simulate DDoS attacks to test our mitigation systems
- As an IT Administrator, I need to manage user access levels to ensure proper security
- As a Viewer, I want to see defense effectiveness reports to understand our security posture

# 11 Design Diagrams

## 11.1 Context Diagram

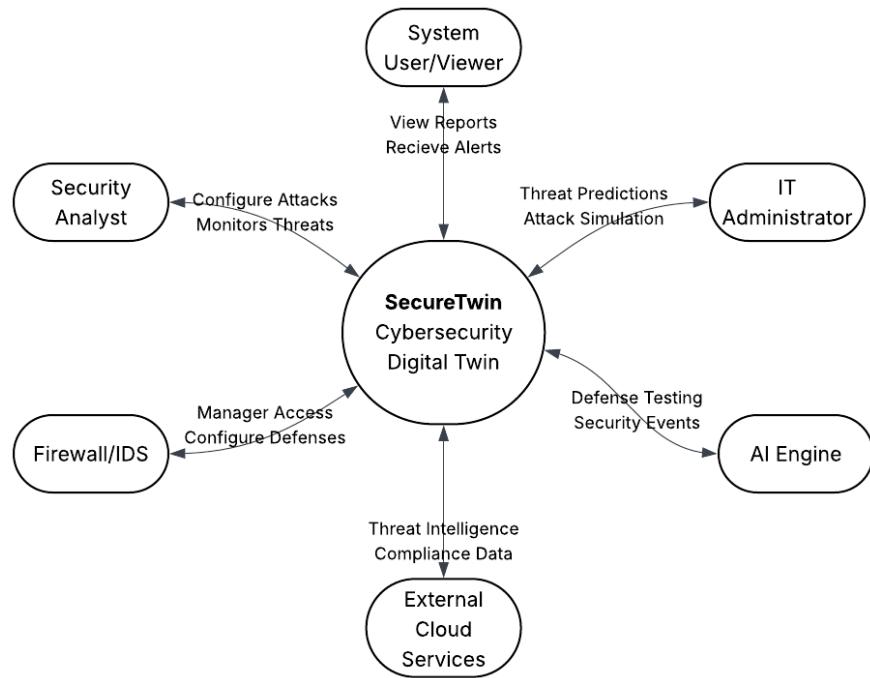


Figure 5: System Context Diagram

## 11.2 Level 1 Diagram

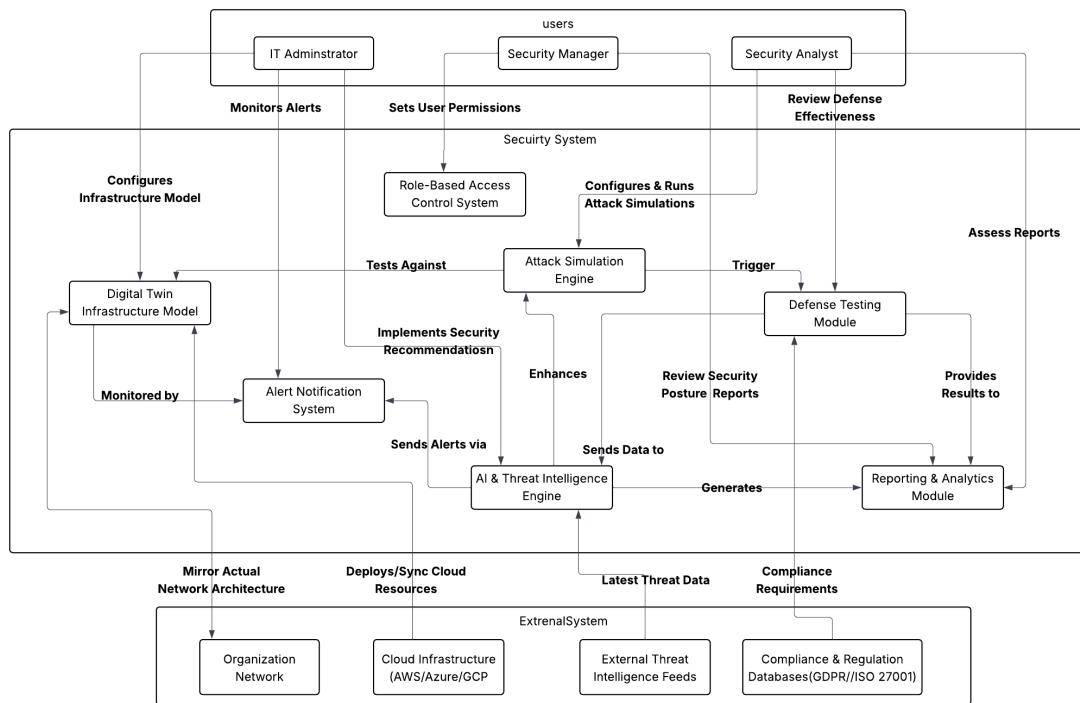


Figure 6: Level 1 Diagram for SecureTwin

## 11.3 UML Class Diagram

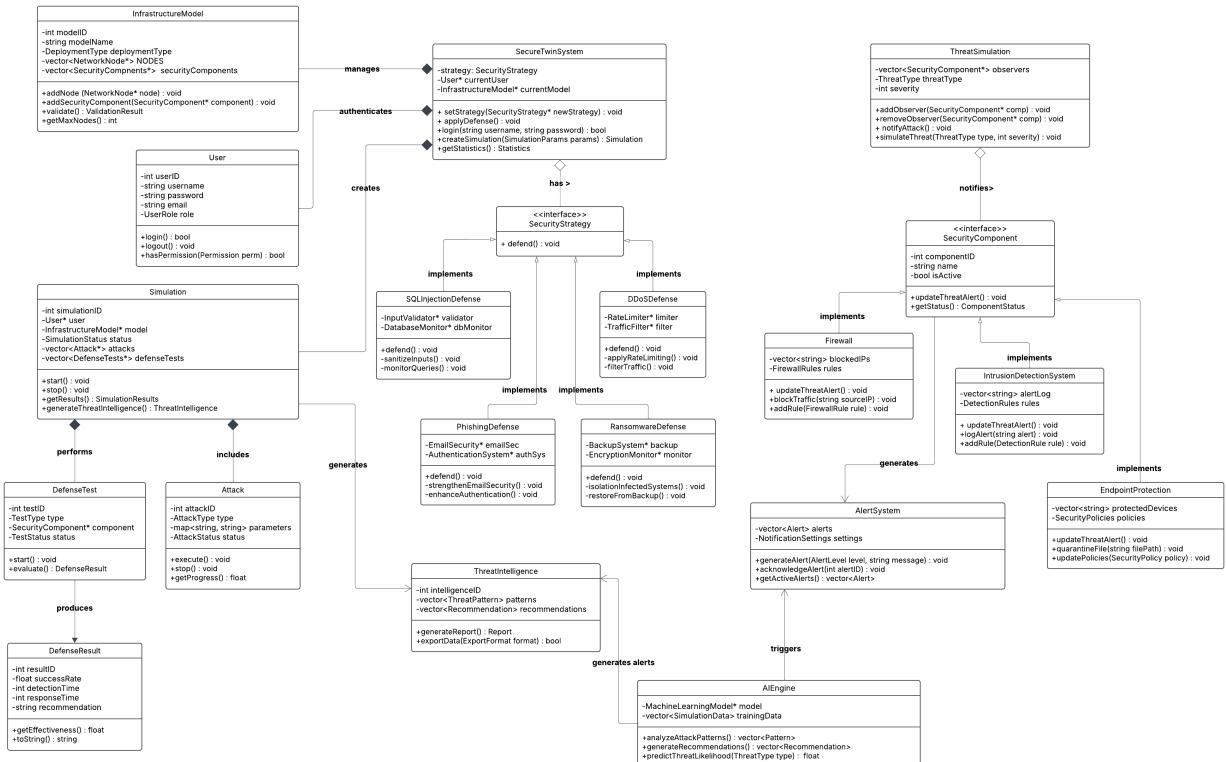


Figure 7: SecureTwin Class Diagram

## Design Patterns in UML

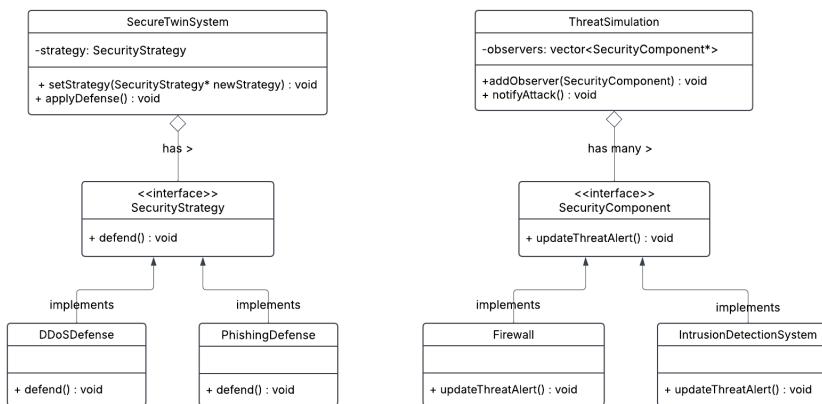


Figure 8: Highlighted Strategy and Observer Patterns in SecureTwin Architecture

## 11.4 ER Diagram

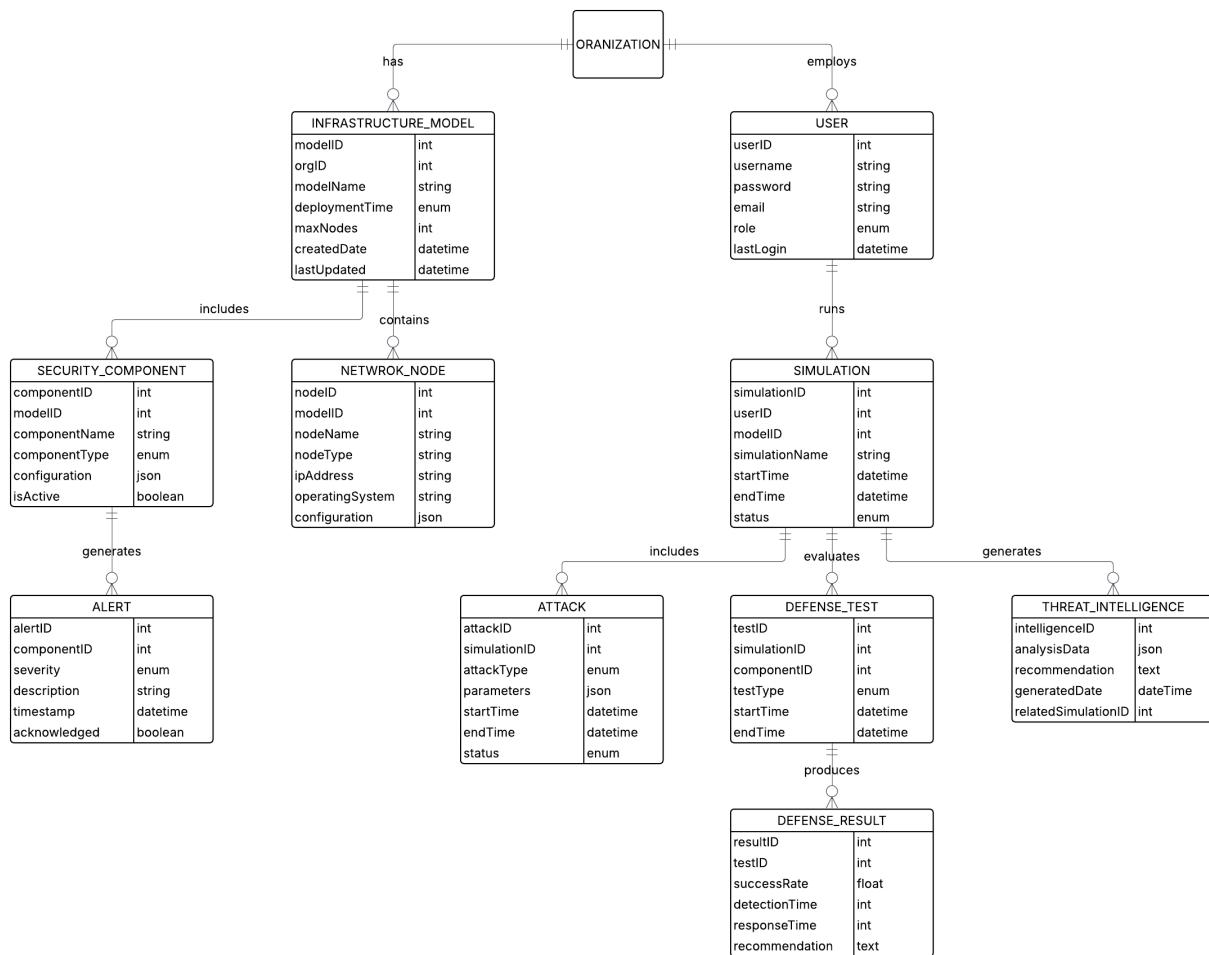


Figure 9: Database Entity-Relationship Diagram

## 12 System Requirements

### 12.1 Hardware Requirements

Component	Specification
Development Workstation	<ul style="list-style-type: none"> <li>• CPU: Intel i7 or equivalent (8 cores minimum)</li> <li>• RAM: 32GB DDR4</li> <li>• Storage: 1TB SSD + 2TB HDD</li> <li>• GPU: NVIDIA RTX 3060 or equivalent</li> </ul>
Production Server	<ul style="list-style-type: none"> <li>• CPU: Dual Xeon Silver 4210 (20 cores total)</li> <li>• RAM: 128GB ECC DDR4</li> <li>• Storage: RAID 10 configuration (4x 2TB SSD)</li> <li>• Network: 10Gbps Ethernet</li> </ul>
Testing Environment	<ul style="list-style-type: none"> <li>• Minimum 5 physical nodes for network simulation</li> <li>• IoT devices for attack surface testing</li> </ul>

Table 5: Hardware Specifications

### 12.2 Software Requirements

Category	Requirements
Operating Systems	<ul style="list-style-type: none"> <li>• Development: Windows 10/11 or Linux (Ubuntu 20.04+)</li> <li>• Production: CentOS 8 or Ubuntu Server 20.04 LTS</li> </ul>
Development Tools	<ul style="list-style-type: none"> <li>• Python 3.9+, C++17 compiler</li> <li>• Node.js 16.x, React 18+</li> <li>• Docker 20.10+, Kubernetes 1.23+</li> </ul>
AI/ML Frameworks	<ul style="list-style-type: none"> <li>• TensorFlow 2.8+ or PyTorch 1.12+</li> <li>• Scikit-learn 1.0+</li> </ul>
Security Tools	<ul style="list-style-type: none"> <li>• Nmap 7.90+, Metasploit Framework</li> <li>• Wireshark 3.6+</li> </ul>

Table 6: Software Specifications

## 13 UI/UX Design

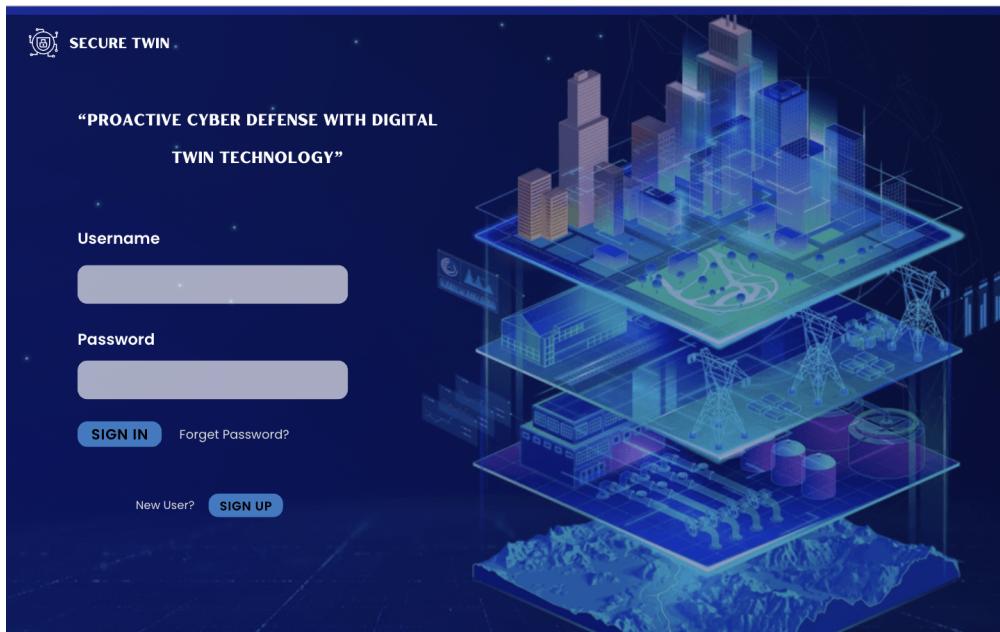


Figure 10: User Login Page (Authentication Flow)

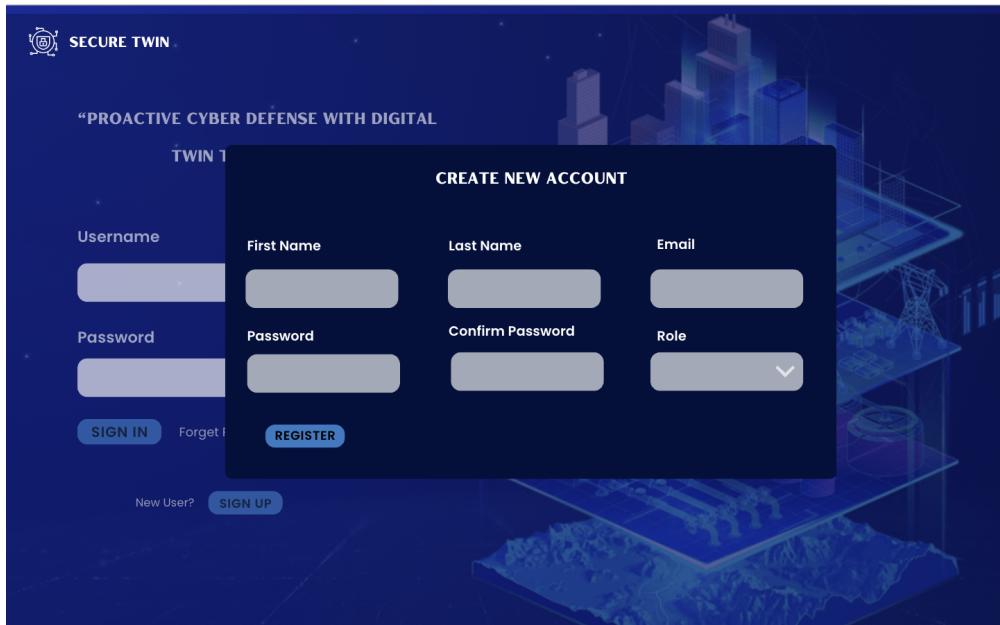


Figure 11: Account Creation Page (Role-Based Registration)

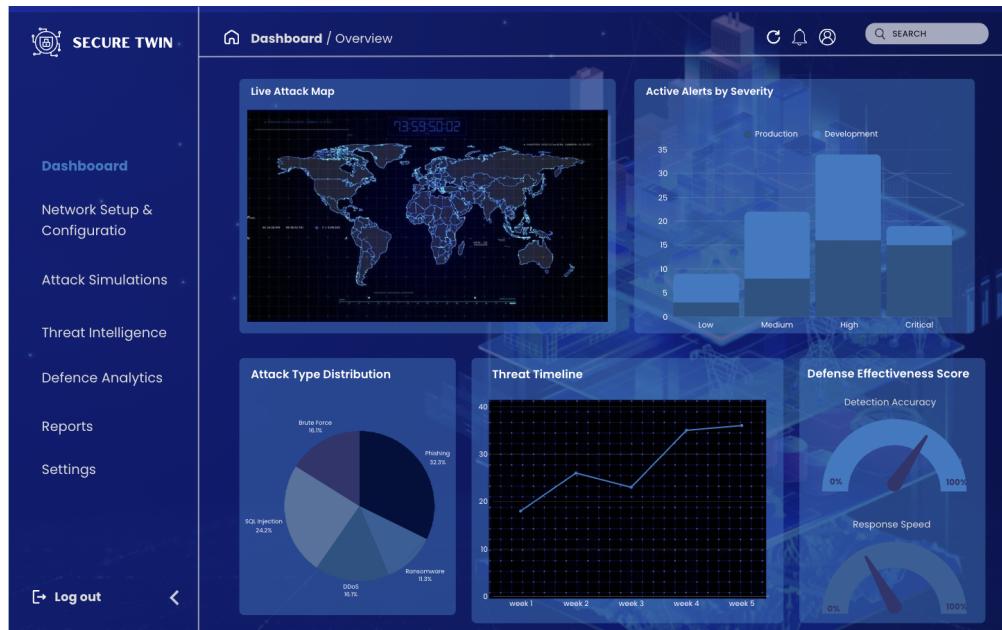


Figure 12: Main Dashboard (Real-Time Security Overview)

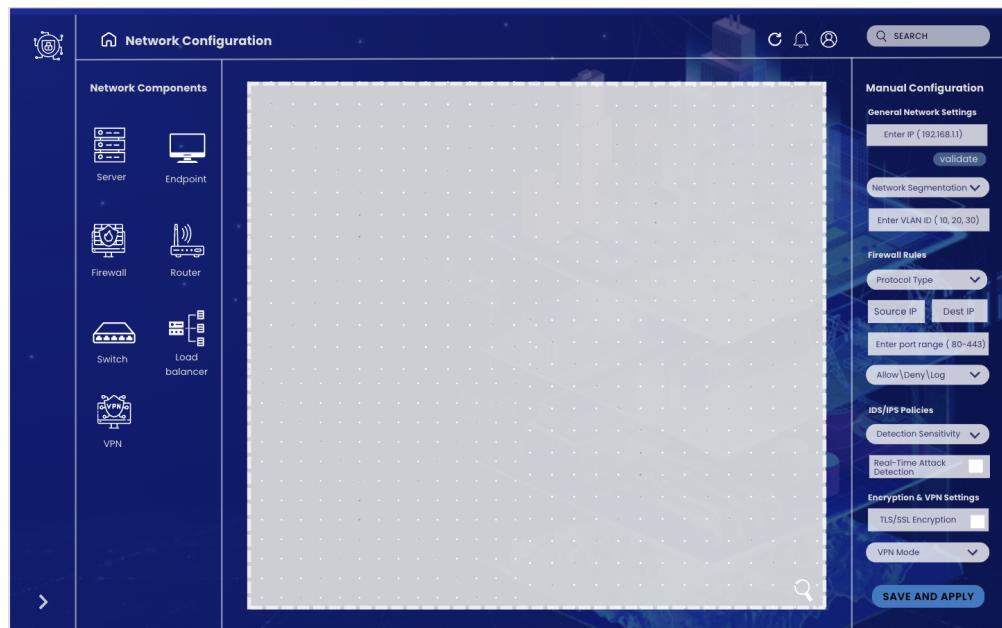


Figure 13: Network Configuration Interface

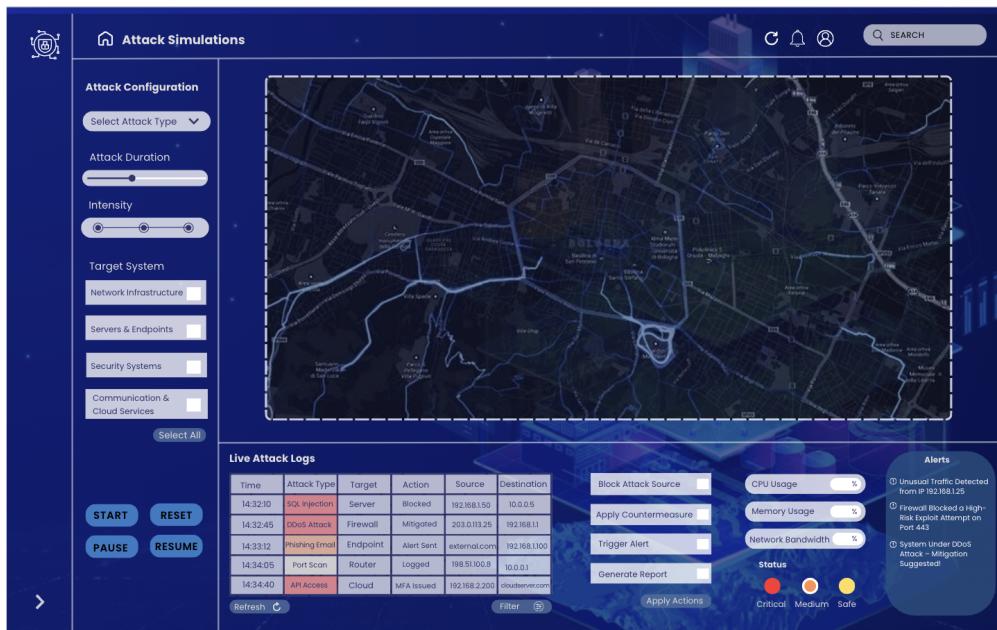


Figure 14: Attack Simulation Panel

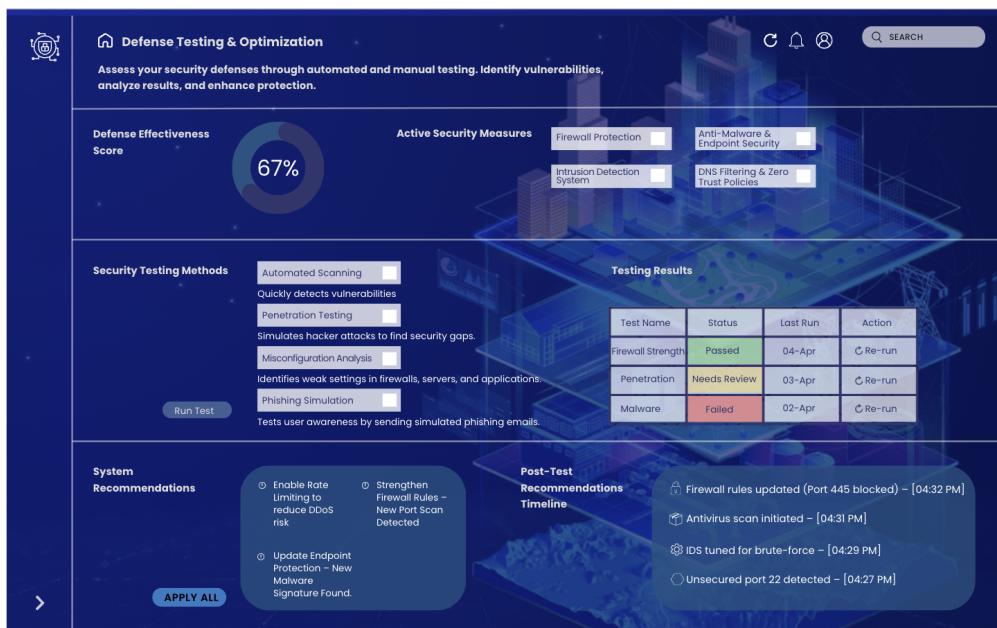


Figure 15: Defense Testing Interface

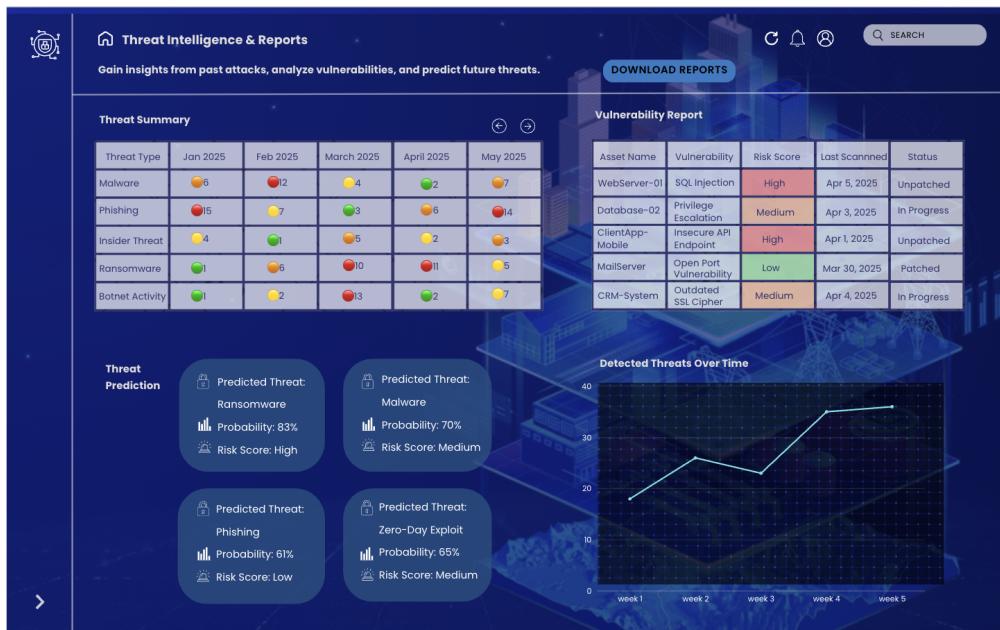


Figure 16: Threat Intelligence Dashboard

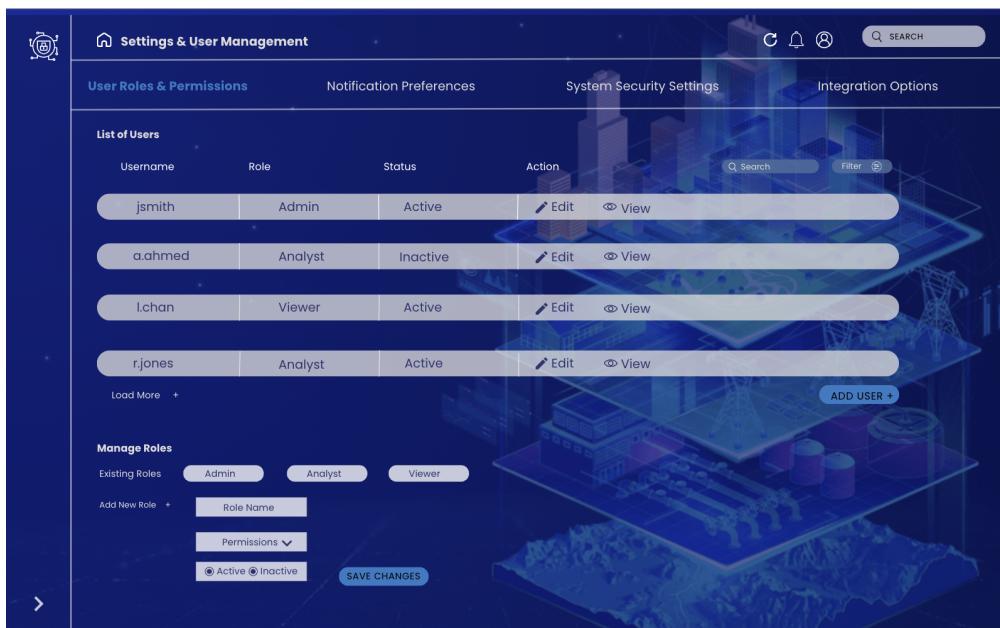


Figure 17: User Management Console

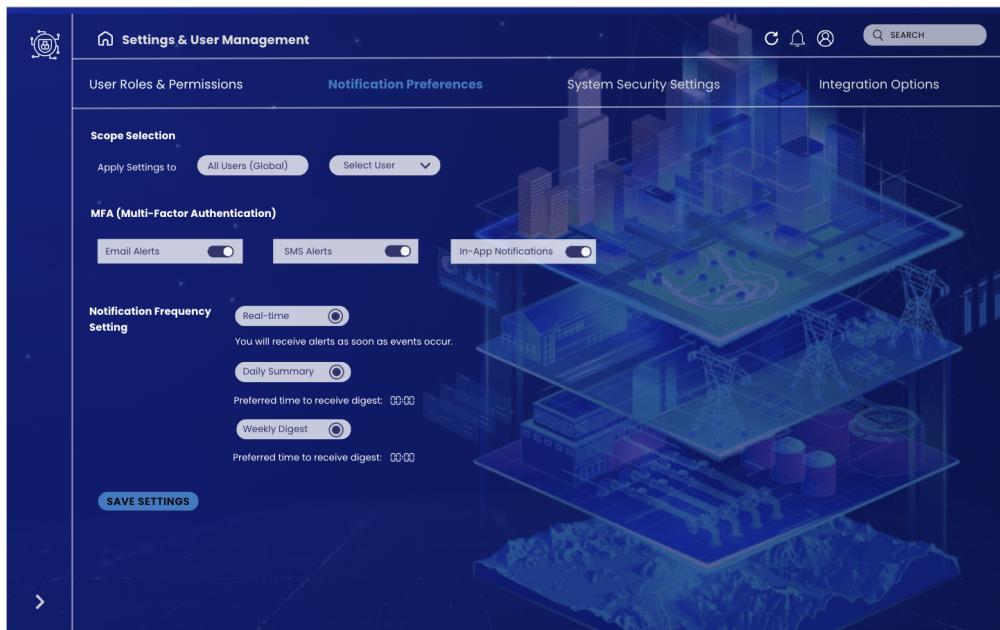


Figure 18: Notification Preferences

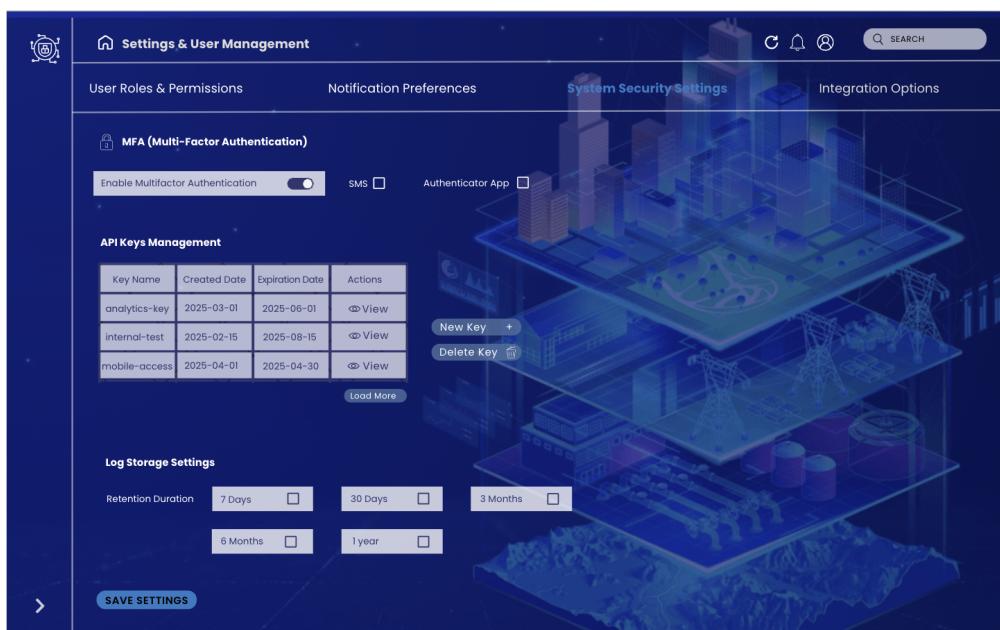


Figure 19: Security Settings

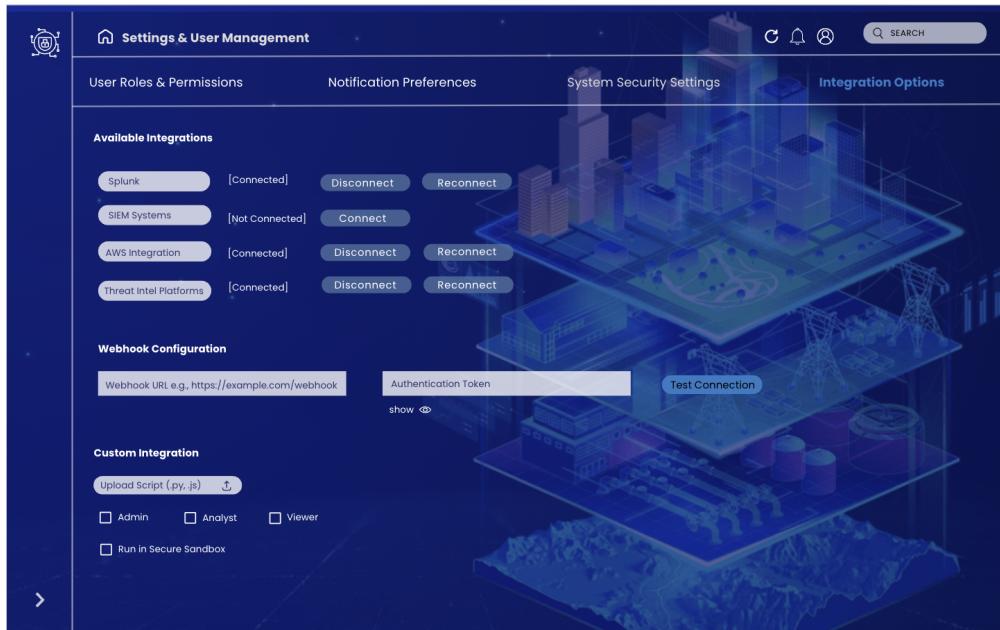


Figure 20: Integration Options

## 14 Conclusion

SecureTwin provides a comprehensive solution for proactive cybersecurity testing through digital twin technology. The project has demonstrated:

- Strong market potential with competitive differentiation
- Technical feasibility through architecture validation
- Clear requirements traceable to use cases
- Robust design patterns for scalability
- Comprehensive project planning

Future work will focus on expanding attack simulation scenarios and integrating with additional security platforms.

## References

- [1] Gartner. (2024). *Market Guide for Cybersecurity Simulation*. Gartner Research.
- [2] International Organization for Standardization. (2022). *ISO/IEC 27001:2022 Information security management systems*. ISO.
- [3] European Union. (2018). *General Data Protection Regulation (GDPR)*. Official Journal of the European Union.

- [4] Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
- [5] Shostack, A. (2014). *Threat Modeling: Designing for Security*. Wiley.
- [6] Saudi Vision 2030. (2021). *National Cybersecurity Strategy*. Kingdom of Saudi Arabia.
- [7] National Institute of Standards and Technology. (2023). *NIST Cybersecurity Framework (CSF) 2.0*. U.S. Department of Commerce.
- [8] Tao, F., & Qi, Q. (2023). *Digital Twin Driven Smart Manufacturing*. Academic Press.
- [9] IBM Security. (2024). *AI in Cybersecurity: Threat Detection and Response*. IBM X-Force Threat Intelligence Index.
- [10] React Three Fiber Documentation. (2024). *3D Visualization for React Applications*. <https://docs.pmnd.rs/react-three-fiber>
- [11] Paszke, A., et al. (2023). *PyTorch: An Imperative Style High-Performance Deep Learning Library*. NeurIPS.