ASM-Enterprise: Threat-Intelligent Attack Surface Management Platform

Phase 2 - Main Project Proposal

Project Overview

Project Title: ASM-Enterprise - Real-time Threat Intelligence Platform
Duration: 20 weeks (10 Scrum sprints × 2 weeks each)
Core Purpose: Enterprise-grade attack surface management with automated threat correlation and MITRE ATT&CK framework integration

What makes this enterprise-grade:

- **Real-time threat intelligence** - Integrates with CVE databases and threat feeds
- **MITRE ATT&CK mapping** - Correlates discovered assets with known attack techniques
- **Scalable architecture** - Handles thousands of assets with cloud deployment
- **Executive dashboards** - Management-friendly visualizations and reports

---

Technical Architecture

Technology Stack

```
Backend: Python FastAPI with async processing
Database: PostgreSQL (enterprise-grade)
Frontend: Plotly Dash for interactive dashboards
Message Queue: Redis for task processing
Threat Intel: STIX 2.0 format integration
Cloud: Azure Container Registry + App Service
CI/CD: Jenkins with automated security scanning
```

Core Components

1. Asset Discovery Engine

- Continuous monitoring of internet-facing assets
- Multi-source data correlation
- Automated asset classification

2. Threat Intelligence Correlation

- Real-time CVE database integration

- MITRE ATT&CK technique mapping
- Custom threat scoring algorithms

3. Visualization Dashboard

- Executive summary views
- Technical deep-dive interfaces
- Risk trending and analytics

4. Automated Reporting

- NIST CSF 2.0 compliance reports
- Stakeholder-specific dashboards
- Incident response playbooks

Advanced Scrum Framework

Enhanced Team Structure

- **External Product Owner:** Industry security expert (validates real-world needs)
- **Scrum Master:** Faculty + backup technical lead
- **Development Team:** You (student)
- **Validation Partners:** 3 organizations for beta testing

Enterprise Scrum Book

More comprehensive than Phase 1:

Section 1: Product Backlog

- Epic-level planning with JIRA integration
- Industry stakeholder prioritization
- Sprint retrospectives with lessons learned

Section 2: Database & UI Design

- PostgreSQL schema evolution
- Dashboard wireframes and user flows
- API documentation

Section 3: Testing & Security

- Automated testing with Selenium
- Security scanning with OWASP ZAP

- Penetration testing reports

Section 4: DevOps & Deployment

- CI/CD pipeline configurations
- Docker container management
- Cloud deployment manifests

💡 *Tip: This book becomes your professional portfolio artifact*

Stakeholder Engagement Strategy

Industry Validation Program

Week 3: University Security Operations Center (SOC)

- Penetration testing requirements workshop
- Get signed MOU for testing environment

Week 6: Industry Partner (CMMI Level 3 company)

- API usability testing session
- Video recording of feedback session

Week 10: NIST Framework Consultant

- CSF 2.0 compliance review
- Formal audit report

Week 15: Open-source Community

- GitHub code review session
- Pull request feedback integration

Week 18: Azure Cloud Architect

- Cloud optimization consultation
- Cost analysis and scaling recommendations

Professional Documentation

- Signed engagement letters
- Video conferences with transcripts
- Professional email chains
- LinkedIn endorsements from participants

Development Timeline

Sprint Planning (20 weeks)

Sprints 1-2 (Weeks 1-4): Foundation

- Infrastructure setup and CI/CD pipeline
- Core database schema
- Basic threat intelligence integration

Sprints 3-4 (Weeks 5-8): Discovery Engine

- Advanced asset discovery algorithms
- Multi-source data correlation
- Performance optimization for large datasets

Sprints 5-6 (Weeks 9-12): Threat Intelligence

- CVE database integration
- MITRE ATT&CK framework mapping
- Custom threat scoring algorithms

Sprints 7-8 (Weeks 13-16): Visualization

- Executive dashboard development
- Interactive threat landscape views
- Report generation engine

Sprints 9-10 (Weeks 17-20): Enterprise Deploy

- Cloud deployment and scaling
- Load testing and performance tuning
- Final security hardening

Key Milestones

- **Week 4:** NVD integration with 90% CVE accuracy
- **Week 8:** Continuous monitoring under 5 minutes per scan
- **Week 12:** Executive dashboard with real-time updates
- **Week 16:** Load testing with 10,000+ assets
- **Week 20:** Live Azure deployment with demo

Testing & Quality Assurance

Test-Driven Development (TDD)

python

```python
# Example enterprise-level tests
def test_cve_correlation_accuracy():
    """Test threat intelligence correlation"""
    test_cases = [
        ("CVE-2024-1234", 7.5, "High"),
        ("CVE-2024-5678", 3.2, "Low")
    ]
    for cve, score, expected in test_cases:
        assert categorize_threat(cve, score) == expected

def test_mitre_attack_mapping():
    """Test MITRE ATT&CK integration"""
    technique = "T1190"  # Exploit Public-Facing Application
    assert get_mitre_tactic(technique) == "Initial Access"

def test_scalability_performance():
    """Test system performance under load"""
    assets = generate_test_assets(10000)
    start_time = time.time()
    results = process_asset_batch(assets)
    duration = time.time() - start_time
    assert duration < 300  # Must complete within 5 minutes
```

Security Testing Requirements

- OWASP ZAP automated scanning
- Bandit security linting
- Container vulnerability scanning
- Penetration testing by university SOC

Cloud Deployment Strategy

Azure Architecture

yaml

```yaml
# Azure Resource Configuration
services:
  container_registry:
    - name: asmenterpriseacr
    - sku: Standard
    - location: East US

  app_service:
    - name: asm-enterprise-app
    - runtime: Python 3.10
    - auto_scaling: enabled
    - ssl_certificate: Let's Encrypt

  database:
    - service: Azure Database for PostgreSQL
    - tier: General Purpose
    - backup_retention: 7 days
```

DevOps Pipeline

```yaml
# Jenkins Pipeline Configuration
pipeline:
  stages:
    - name: Build
      commands:
        - docker build -t asm-enterprise:$BUILD_NUMBER .
        - docker scan asm-enterprise:$BUILD_NUMBER

    - name: Test
      commands:
        - pytest --cov=src --cov-report=xml
        - bandit -r src -f xml -o security_scan.xml
        - owasp-zap-baseline.py -t http://localhost:8000

    - name: LaTeX
      commands:
        - pdflatex -interaction=nonstopmode report/main.tex
        - bibtex report/main
        - pdflatex -interaction=nonstopmode report/main.tex

    - name: Deploy
      conditions: branch == 'main'
      commands:
        - az acr login --name asmenterpriseacr
        - docker push asmenterpriseacr.azurecr.io/asm-enterprise:$BUILD_NUMBER
        - az webapp deployment source config --name asm-enterprise-app
```

Compliance & Deliverables

Academic Requirements

- [ ] Scrum Book with 300+ dated entries
- [ ] Git repository with 120+ semantic commits
- [ ] LaTeX report (40 pages, ACM SIGS format)
- [ ] Industry validation with signed assessments
- [ ] Live deployment with public URL

Professional Deliverables

- [ ] Enterprise-grade software architecture
- [ ] Cloud deployment manifests
- [ ] Security assessment reports
- [ ] Load testing documentation

- [ ] API documentation with OpenAPI spec

Publication Opportunities

- [ ] IEEE Security & Privacy conference abstract
- [ ] Open-source project with community adoption
- [ ] Industry case study publication
- [ ] Professional certification pathway

Risk Management

Enterprise-Level Risks

Risk: Cloud cost overrun
Mitigation: Auto-shutdown scripts (9PM-7AM), budget alerts at ₹7,500
Trigger: Daily cost monitoring dashboard

Risk: Third-party API downtime
Mitigation: Local NVD mirror, fallback data sources
Trigger: 3 consecutive API failures

Risk: Stakeholder schedule conflicts
Mitigation: Dual-track development, async communication
Trigger: 48-hour response time exceeded

Risk: Scaling performance issues
Mitigation: Microservices architecture, load testing
Trigger: Response time >5 seconds

Resource Planning

Infrastructure Requirements

yaml

```yaml
development:
  hardware:
    - cpu: i7 processor (minimum)
    - memory: 16GB RAM
    - storage: 256GB SSD

  cloud:
    - azure_credits: ₹8,000 budget
    - services: ACR + App Service + PostgreSQL
    - monitoring: Application Insights

software_licenses:
  - jenkins: Open source
  - owasp_zap: Open source
  - nessus: Academic license
  - azure_devops: Student subscription
```

Time Investment

- **Baseline:** 18 hours/week
- **Peak periods:** 25 hours/week (Weeks 10-16)
- **Buffer:** 20% timeline reserve in final sprints

Success Metrics

Technical Metrics

- **Performance:** <5 minutes for 10,000 asset scan
- **Accuracy:** 90%+ CVE correlation accuracy
- **Scalability:** Support for 50,000+ assets
- **Uptime:** 99.5% availability on Azure

Academic Metrics

- **Compliance:** 99% syllabus requirement fulfillment
- **Innovation:** Novel threat scoring algorithm
- **Research:** Accepted conference/journal submission
- **Validation:** 5+ industry stakeholder endorsements

Professional Metrics

- **Portfolio:** Enterprise-grade reference project

- **Skills:** Cloud architecture and DevOps proficiency
- **Network:** 10+ industry professional connections
- **Career:** Job placement pipeline with participating companies

---

Getting Started

Pre-Phase 2 Requirements

- [ ] Successful completion of Phase 1 (ASM-Lite)
- [ ] Faculty approval for enterprise scope
- [ ] Azure student subscription activated
- [ ] Industry mentor identified
- [ ] Security clearance (if working with sensitive data)

Week 1 Action Items

1. Set up Azure environment and cost monitoring
2. Establish industry partnerships
3. Configure enterprise development tools
4. Create detailed technical architecture
5. Initialize enterprise Scrum Book
6. Schedule industry stakeholder meetings

Faculty Checkpoint

- [ ] Technical architecture review
- [ ] Resource allocation approval
- [ ] Industry partnership validation
- [ ] Timeline and milestone agreement
- [ ] Risk management plan approval

Note: This is a significant step up from Phase 1. The enterprise focus means real-world validation and professional-grade deliverables. Consider this your capstone project that bridges academic learning with industry practice.