1. Phishing Detection and Awareness Platform

Why This Project?

- Addresses the 42% increase in phishing attacks reported in 2024 https://cybermagazine.com/articles/netskope-data-shows-phishing-success-rate-tripled-in-2024
- https://www.infosecurity-magazine.com/news/phishing-click-rates-triple/
- https://www.fortinet.com/products/fortimail-workspace-security
- Combines machine learning, web development, and cybersecurity fundamentals
- Scalable complexity - can start simple and add advanced features

Team Role Distribution

1. Threat Intelligence Analyst

Responsibilities:

- Research current phishing trends and attack vectors
- Analyse threat feeds from PhishTank, VirusTotal, and URLVoid
- Document emerging AI-powered phishing techniques
- Coordinate with team members on threat landscape updates

Skills Developed:

- Threat hunting and intelligence gathering
- OSINT (Open Source Intelligence) techniques
- Incident response coordination
- Risk assessment and documentation

Tools & Resources:

- PhishTank API for real-time phishing URLs
- MITRE ATT&CK framework for attack classification
- Threat intelligence platforms (OpenCTI, MISP)

2. Machine Learning Security Engineer (enhanced ML Specialist)

Responsibilities:

- Develop advanced classification models using Random Forest, Neural Networks, and NLP
- Implement behavioural analysis for user interaction patterns
- Create AI-powered detection systems for zero-day phishing attempts
- Build automated feature extraction for URL, email, and content analysis

Skills Developed:

- Cybersecurity-focused machine learning
- Feature engineering for security datasets
- Model deployment and monitoring
- AI threat detection techniques

Enhanced Datasets & Tools:

- Primary Dataset: PhishTank + Alexa Top Sites (10,000+ samples)
- Advanced Features: URL lexical analysis, WHOIS data, SSL certificate validation
- ML Frameworks: TensorFlow, scikit-learn, NLTK/SpaCy for NLP
- Security Libraries: Python security modules, API integrations

3. Cybersecurity Full-Stack Developer (enhanced Web Developer)

Responsibilities:

- Build secure web platform with authentication and authorization
- Implement real-time threat visualization dashboards
- Create phishing simulation training modules
- Develop API endpoints for threat intelligence integration

Skills Developed:

- Security-first web development practices
- Secure coding principles (OWASP Top 10)
- Real-time data visualization for security operations
- API security and authentication

Technology Stack:

- Frontend: React.js with security headers, CSP implementation

- Backend: Python Flask/Django with security middleware
- Database: PostgreSQL with encryption at rest
- Visualization: D3.js, Chart.js for threat intelligence dashboards
- Security: JWT authentication, HTTPS enforcement, input validation

4. Security Data Scientist (enhanced Data Analyst)

Responsibilities:

- Perform statistical analysis on phishing campaign effectiveness
- Evaluate model performance using cybersecurity-specific metrics
- Analyze user behavior patterns and click-through rates
- Generate threat intelligence reports and visualizations

Skills Developed:

- Security analytics and metrics
- Statistical analysis of cyber threats
- Performance evaluation of security controls
- Data-driven security decision making

Analysis Focus:

- Model Metrics: Precision, Recall, F1-Score, ROC-AUC for imbalanced datasets
- Security KPIs: False positive rates, detection time, user susceptibility analysis
- Trend Analysis: Attack vector evolution, seasonal patterns, industry targeting

5. Penetration Testing & Social Engineering Specialist (enhanced Security Researcher)

Responsibilities:

- Conduct ethical phishing simulations using Gophish framework
- Research social engineering techniques and psychological triggers
- Test platform security through controlled penetration testing
- Develop awareness training content based on real-world scenarios

Skills Developed:

- Ethical hacking and penetration testing
- Social engineering assessment techniques
- Security awareness training development
- Vulnerability assessment and remediation

Tools & Techniques:

- Phishing Frameworks: Gophish, KingPhish, Phishing Frenzy
- Testing Tools: Metasploit, BeEF, ZAP for security assessment
- Training Platforms: Custom simulation environments
- Research Methods: Analysis of current phishing campaigns and techniques
-

Free Tools and Resources:

- Datasets: PhishTank, UCI Machine Learning Repository
- Development: Python, Visual Studio Code, GitHub
- Deployment: Heroku free tier or GitHub Pages

Tasks per week according to role

Week	Threat Intelligence Analyst (Role 1) Pabin	ML Security Engineer (Role 2) Pramit	Full-Stack Developer (Role 3) Bishal	Security Data Scientist (Role 4) Sadaiba	Pen Testing & Social Engineering (Role 5) Roshan
1	Research latest phishing trends, set up threat sources (PhishTank, VirusTotal); prep threat intel doc	Prepare ML project structure, gather initial datasets (download PhishTank, Alexa, etc.)	Plan overall platform architecture; set up repository structure	Collect data for exploratory analysis; basic data cleaning	Review common phishing/social engineering techniques; propose testing plan
2	Analyze collected threat data; summarize major attack vectors; share with team	Clean and preprocess datasets; develop feature extraction scripts	Build project skeleton: setup backend, basic frontend	Analyze dataset statistics; visualize data patterns	Install/configure Gophish for local testing; draft initial training scenarios

3	Identify emerging AI-powered phishing tactics; gather OSINT resources	Implement baseline ML models (Random Forest, etc.); run initial tests	Implement authentication and authorization endpoints	Begin evaluation metrics setup (recall, precision); analyze feature importances	Run internal phishing simulations; document staff/user response mechanisms
4	Integrate live threat feeds/automation; update MITRE ATT&CK mapping	Experiment with advanced models (NLP, deep learning); begin behavioral analysis	Develop REST API endpoints for threat data; connect backend to database	Track early model outputs; visualize user behavior data	Test web app security for common vulnerabilities (OWASP); initial findings report
5	Document risk assessments; coordinate team updates on new threats	Fine-tune feature engineering (SSL, WHOIS, lexical analysis)	Implement dashboard UI for viewing alerts/threats	Analyze phishing campaign effectiveness over time, summarize findings	Update phishing simulation scenarios with new techniques
6	Monitor threat landscape for changes; review/merge risk documentation	Integrate model with backend for real-time detection	Add real-time threat visualization (charts, alerts)	Compare results from multiple models;	Pen-test platform authentication/API endpoints; report vulnerabilities

				improve visualizations	
7	Prepare threat intelligence summary for mid-term review; refine docs	Evaluate detection model performance (on pre-processed/test sets)	Extend secure login/session handling; harden backend security	Statistical analysis on detection rates; report KPIs	Deliver phishing/social engineering awareness training session to team
8	Update AI-phishing technique database; review competitor platforms	Deploy AI detection as API, run cross-validation	Build simulation module for phishing-awareness training	Generate trend analysis by attack type/source	Simulate advanced phishing campaigns; gather feedback on realism
9	Lead threat briefing on new campaign methods; revise risk docs	Optimize model speed and accuracy; test on real-world data	Integrate simulation results to user dashboard	Finalize performance evaluation; write summary for doc	Conduct live penetration tests on whole app; summarize risks/remediations
10	Coordinate incident response documentation; consolidate lessons	Support testing of deployed ML models; monitor logs	Refactor UI/UX; user preference and accessibility improvements	Document user susceptibility findings;	Prepare final awareness/training materials for user education module

				refine visual trends	
11	Prepare documentation for assessment; update with latest threats	Review/cleanup ML scripts; prepare evaluation/usage guide	Finalize dashboard/reporting; cleanup codebase	Produce final analytics charts/graphs for report	Complete penetration testing documentation; work on overall security report
12	Cross-check project for threat/attack coverage; assist teammates	Help test integration in “production” environment	Help doc team with user/deployment instructions	Final checks on all evaluation data; update graphs/tables	Support team in any last-minute platform/awareness tasks as needed
13	Polish final threat docs for presentation/report	Refine model visuals/results for demo	Assist in demo prep; run platform walkthrough	Prepare data for presentation slides; answer data-driven questions	Demo phishing simulation & live scenario for stakeholders
14	Present role highlights and team findings	Present ML techniques and lessons learned	Present platform features and security wins	Present key analytics & project outcomes	Present social engineering lessons and real hacking demos

Comprehensive 11-Week Timeline

Week 1: August 1-7 (Project Foundation)

All Team Members - Setup Tasks

Jira Tasks:

- PROJ-1: Create Jira project and configure Kanban board
- PROJ-2: Set up team permissions and project roles
- PROJ-3: Create initial epics and project roadmap

Bitbucket Tasks:

- Initialize repository with proper branch structure
- Configure Jira integration and smart commits
- Set up branch permissions and pull request workflows
- Create initial README and project documentation

Weekly Deliverables:

- Fully configured Jira project with team access
- Bitbucket repository with proper branching strategy
- Jira-Bitbucket integration working with test commits

Individual Team Member Tasks:

Threat Intelligence Analyst

Jira Tasks:

- PHISH-1: Create comprehensive project charter
- PHISH-2: Research 2024 phishing attack trends and statistics
- PHISH-3: Identify and document threat intelligence API sources

Bitbucket Work:

- Create `feature/PHISH-1-project-charter` branch
- Push project documentation to `/docs` folder
- Create threat intelligence source documentation

ML Security Engineer

Jira Tasks:

- PHISH-4: Set up ML development environment
- PHISH-5: Design dataset collection and processing strategy
- PHISH-6: Plan feature engineering approach for phishing detection

Bitbucket Work:

- Create `feature/PHISH-4-ml-environment` branch
- Set up `requirements.txt` with ML dependencies
- Create initial ML pipeline structure in `/ml` folder

Cybersecurity Full-Stack Developer

Jira Tasks:

- PHISH-7: Design system architecture and component interaction
- PHISH-8: Select and document technology stack
- PHISH-9: Create security requirements and OWASP compliance checklist

Bitbucket Work:

- Create `feature/PHISH-7-architecture` branch
- Set up project folder structure
- Create security configuration templates

Week 2: August 8-14 (Core Foundation)

Threat Intelligence Analyst

Jira Tasks:

- PHISH-10: Implement threat feed API integrations (PhishTank, VirusTotal, URLVoid)
- PHISH-11: Create first weekly threat intelligence briefing

- PHISH-12: Establish team coordination and communication protocols

Bitbucket Work:

- Branch: `feature/PHISH-10-threat-feeds`
- Implement API integration modules
- Create automated threat data collection scripts
- Update documentation with API usage examples

ML Security Engineer

Jira Tasks:

- PHISH-13: Collect and validate phishing datasets (5,000+ samples)
- PHISH-14: Perform exploratory data analysis and visualization
- PHISH-15: Build initial ML pipeline with data preprocessing

Bitbucket Work:

- Branch: `feature/PHISH-13-dataset-collection`
- Create data collection and cleaning scripts
- Jupyter notebooks for exploratory data analysis
- Initial ML pipeline with preprocessing modules

Security Data Scientist

Jira Tasks:

- PHISH-16: Design statistical analysis framework
- PHISH-17: Validate dataset quality and completeness
- PHISH-18: Create KPI dashboard mockups and metrics definition

Bitbucket Work:

- Branch: `feature/PHISH-16-analytics-framework`
- Statistical analysis modules and utilities
- Data validation scripts and quality reports
- Dashboard design files and prototypes

Penetration Testing Specialist

Jira Tasks:

- PHISH-19: Research current social engineering techniques
- PHISH-20: Install and configure Gophish framework
- PHISH-21: Design 5 realistic phishing simulation scenarios

Bitbucket Work:

- Branch: `feature/PHISH-19-social-engineering`
- Gophish configuration files and setup scripts
- Phishing simulation templates and scenarios
- Documentation for ethical testing procedures

Week 3: August 15-21 (Development Acceleration)

ML Security Engineer

Jira Tasks:

- PHISH-22: Implement advanced feature engineering (URL, email, content analysis)
- PHISH-23: Train initial ML models (Random Forest, Logistic Regression, Neural Networks)
- PHISH-24: Evaluate and compare model performance metrics

Bitbucket Work:

- Branch: `feature/PHISH-22-feature-engineering`
- Feature extraction modules with 20+ security-focused features
- Model training scripts with cross-validation
- Performance evaluation and comparison reports

Cybersecurity Full-Stack Developer

Jira Tasks:

- PHISH-25: Develop RESTful API with Flask/Django backend
- PHISH-26: Implement database schema with security considerations
- PHISH-27: Create JWT-based authentication and authorization system

Bitbucket Work:

- Branch: `feature/PHISH-25-backend-api`

- API endpoints for threat intelligence and ML predictions
- Secure database models with encryption
- Authentication middleware and security headers

Threat Intelligence Analyst

Jira Tasks:

- PHISH-28: Integrate real-time threat feeds with platform
- PHISH-29: Implement automated threat classification system
- PHISH-30: Create threat scoring algorithm

Bitbucket Work:

- Branch: `feature/PHISH-28-realtime-integration`
- Real-time threat feed processing
- Threat classification and scoring modules
- Integration with ML pipeline for enhanced detection

Week 4: August 22-28 (Integration Phase 1)

Cross-Team Integration Focus

Jira Tasks:

- INTEG-1: Integrate ML models with web platform APIs
- INTEG-2: Connect threat intelligence feeds with detection system
- INTEG-3: Implement real-time data flow between components

Bitbucket Work:

- Branch: `integration/phase1`
- API integration between ML and web components
- Real-time data processing pipeline
- Initial end-to-end testing framework

Team Collaboration:

- Daily standups documented in Jira comments
- Pull request reviews with mandatory approvals from 2 team members
- Integration testing with automated Bitbucket Pipelines

Week 5: August 29 - September 4 (Feature Enhancement)

Cybersecurity Full-Stack Developer

Jira Tasks:

- PHISH-31: Develop React.js frontend with security dashboard
- PHISH-32: Implement training simulation interface
- PHISH-33: Create user management with role-based access control

Bitbucket Work:

- Branch: `feature/PHISH-31-frontend-dashboard`
- Interactive security dashboard with real-time updates
- Training simulation user interface
- User management system with secure session handling

Penetration Testing Specialist

Jira Tasks:

- PHISH-34: Execute first phishing simulation campaign
- PHISH-35: Analyze user susceptibility patterns and behaviors
- PHISH-36: Document social engineering effectiveness metrics

Bitbucket Work:

- Branch: `feature/PHISH-34-simulation-execution`
- Simulation execution scripts and automation
- User interaction tracking and analysis
- Reporting modules for campaign effectiveness

Week 6: September 5-11 (Mid-Project Review)

Major Milestone: Live Demonstration

Jira Tasks:

- REVIEW-1: Prepare comprehensive platform demonstration

- REVIEW-2: Document progress against initial requirements
- REVIEW-3: Risk assessment and mitigation planning

Bitbucket Work:

- Branch: `release/midpoint-demo`
- Stable demo version with all integrated features
- Comprehensive documentation updates
- Performance benchmarking and optimization

Deliverables:

- 30-minute live platform demonstration
- Technical documentation review
- Progress assessment report
- Updated project timeline and risk mitigation

Week 7: September 12-18 (Advanced Features)

ML Security Engineer

Jira Tasks:

- PHISH-37: Implement advanced NLP for email content analysis
- PHISH-38: Add computer vision for suspicious image/logo detection
- PHISH-39: Create model interpretability and explainability features

Bitbucket Work:

- Branch: `feature/PHISH-37-advanced-nlp`
- NLP modules using BERT/transformer models
- Image analysis using OpenCV and deep learning
- Model explainability dashboard with SHAP/LIME

Week 8: September 19-25 (Security Hardening)

Security Focus Across All Teams

Jira Tasks:

- SEC-1: Comprehensive security audit and penetration testing
- SEC-2: OWASP Top 10 compliance verification
- SEC-3: Performance optimization under security constraints

Bitbucket Work:

- Branch: `security/hardening-phase`
- Security audit findings and remediation
- OWASP compliance implementation
- Performance optimization with security maintained

Week 9: September 26 - October 2 (Testing & Validation)

System-Wide Testing

Jira Tasks:

- TEST-1: End-to-end integration testing
- TEST-2: User acceptance testing with sample users
- TEST-3: Performance and load testing

Bitbucket Work:

- Branch: `testing/comprehensive-validation`
- Automated testing suites (unit, integration, e2e)
- User acceptance test scenarios and results
- Performance benchmarking and optimization

Week 10: October 3-9 (Final Integration)

System Finalization

Jira Tasks:

- FINAL-1: Complete system integration and bug fixes
- FINAL-2: Final documentation and deployment guides
- FINAL-3: Presentation preparation and rehearsal

Bitbucket Work:

- Branch: `release/final-preparation`
- Final bug fixes and optimizations
- Complete deployment documentation
- Presentation materials and demos

Week 11: October 10-16 (Project Delivery)

Final Delivery

Jira Tasks:

- DELIVERY-1: Final presentation and project handover
- DELIVERY-2: Code repository finalization
- DELIVERY-3: Project retrospective and lessons learned

Bitbucket Work:

- Branch: `release/v1.0-final`
- Final production-ready code
- Complete documentation package
- Release notes and deployment instructions

Jira & Bitbucket Integration Workflows

Smart Commits Integration

```
bash

# Automatic issue transitions with commits
git commit -m "PHISH-15 #in-progress Implement data
preprocessing pipeline"
git commit -m "PHISH-15 #resolve Fixed data validation issues
#time 3h"


# Linking commits to multiple issues
```

```
git commit -m "PHISH-22 PHISH-23 Add feature extraction for URL analysis"
```

Pull Request Workflow

1. Create feature branch: `feature/PHISH-XX-description`
2. Develop and commit with Jira issue references
3. Push branch and create pull request
4. Automated checks: Bitbucket Pipelines run tests
5. Code review: Minimum 2 approvals required
6. Jira integration: Issues automatically update status
7. Merge to develop: Automatic Jira transition to "Ready for Testing"

Automated Workflows

- Build status updates in Jira issues
- Deployment tracking in Jira with Bitbucket Pipelines
- Automatic issue transitions based on branch/PR status
- Code quality gates with SonarQube integration

Weekly Progress Review Structure

Every Friday 3 PM - Integrated Progress Review

Jira Dashboard Review (15 minutes)

- Sprint/Kanban board status review
- Burndown analysis and velocity tracking
- Issue resolution and blocker identification
- Epic progress toward major deliverables

Bitbucket Activity Review (10 minutes)

- Code commit activity and quality metrics
- Pull request status and review completion
- Branch management and integration health
- Pipeline success rates and deployment status

Integration Status Report (10 minutes)

- Cross-team dependencies and handoffs
- API integration status and testing results
- Data flow validation between components
- Security compliance and audit progress

Risk Assessment & Planning (10 minutes)

- Technical risks and mitigation strategies
- Resource allocation adjustments
- Timeline compliance and adjustment needs
- Next week priority setting and task assignment

Success Metrics with Tool Integration

Jira Metrics

- Issue completion rate: >95% weekly deliverable success
- Cycle time: Average issue resolution < 3 days
- Epic progress: On-track delivery of major features
- Team velocity: Consistent story point completion

Bitbucket Metrics

- Code quality: >90% automated test coverage
- Pull request efficiency: <24 hour review cycles
- Integration success: >95% pipeline success rate
- Branch management: Clean git history with proper merging