Senior Design Progress Report —Process and Logic

Introduction

Project Summary

SecureWebOps is a modular web security platform designed for small organizations that lack dedicated cybersecurity staff and require accessible, easy-to-use security tools. The primary stakeholder for this system is a small business website administrator or IT generalist responsible for maintaining the security of a public-facing website. System requirements and design decisions are additionally informed by industry perspective from Federal Resources Corporation (FRC), providing insight into common security challenges, usability gaps, and workflow expectations observed across organizations.

In its initial release, SecureWebOps provides website vulnerability scanning initiated through a browser extension, allowing the stakeholder to submit a target URL for analysis. The system processes scanning requests through a secure backend service and stores results for later review. Scan findings are presented to the user through a centralized reporting dashboard, enabling the stakeholder to identify and track potential security issues.

Subsequent releases expand system functionality to include encrypted file handling (such as secure PDF uploads and encryption) and phishing detection tools. These features are implemented incrementally to ensure each release delivers usable functionality while maintaining a manageable development scope for a student-led project. By structuring development into defined releases, SecureWebOps provides immediate value while establishing a scalable foundation for a comprehensive security platform.

Our Goal and Vision

The goal of SecureWebOps is to make essential cybersecurity tools accessible to small organizations that lack the budget, time, or specialized staff required to deploy and manage enterprise-grade security solutions. Many small businesses rely on a single website administrator or general IT staff member, making traditional security platforms impractical due to cost, configuration complexity, and ongoing maintenance requirements. Industry insight from Federal Resources Corporation (FRC) further reinforces that these challenges are common across organizations with limited security maturity.

SecureWebOps addresses this gap by unifying common security workflows into a single, easy-to-use platform. These workflows include submitting a website for vulnerability scanning, securely handling sensitive documents through encryption, evaluating content for

phishing indicators, and reviewing security results through a centralized reporting interface. Rather than requiring multiple disconnected tools, SecureWebOps provides a guided process where users can initiate security actions, receive results, and track findings from one system.

The current minimum viable product (MVP) focuses on website vulnerability scanning initiated through a browser extension. This workflow allows a user to submit a target URL, trigger a scan through the backend system, and review scan results in a dashboard. This initial implementation establishes the foundation for future releases that will expand functionality while preserving a consistent and manageable user experience.

Problems SecureWebOps Solves

Small organizations are particularly vulnerable to security threats due to limited budgets and the absence of dedicated cybersecurity personnel. In many cases, website security responsibilities fall to a single administrator or general IT staff member who must manage security alongside other operational tasks. Industry observations from Federal Resources Corporation (FRC) indicate that these constraints are common across organizations with limited security maturity. SecureWebOps is designed to address the following challenges faced by this stakeholder group:

- Limited Access to Practical Web Security Tools—Knowledge, expensive licenses, or dedicated infrastructure. SecureWebOps lowers this barrier by enabling users to initiate website vulnerability scans directly from a browser extension, eliminating the need for complex installations or specialized hardware.
- Inconsistent Handling of Sensitive Information—Small teams often lack standardized processes for protecting sensitive data such as configuration files, uploaded documents, and credentials. SecureWebOps addresses this issue through planned secure file handling workflows, including encryption of uploaded documents, which are introduced incrementally in later releases of the platform.
- Fragmented Security Workflows—Organizations' security, phishing awareness, and reporting. SecureWebOps reduces this complexity by consolidating these security tasks into a single platform, beginning with website vulnerability scanning as the core MVP capability and expanding in future releases.

Why It Matters

SecureWebOps is designed as a multi-tenant platform that supports multiple organizations using the same system while maintaining separation between users and data. Each organization has its own set of users with defined roles and access permissions, allowing administrators to manage security tasks and view reports while standard users may be limited to initiating scans or reviewing assigned results. Industry insight from Federal Resources Corporation (FRC) highlights that role-based access and tenant separation are

essential for organizations managing security across multiple users and teams. This approach ensures that SecureWebOps can be used by multiple companies simultaneously without exposing sensitive information across organizations.

Website vulnerability scanning within SecureWebOps is intended to be performed on systems owned or managed by the user's organization, such as public-facing company websites or internal web applications. The platform does not scan arbitrary third-party websites. Instead, users initiate scans only against approved targets associated with their organization, ensuring ethical use and alignment with real-world security practices.

By providing browser-triggered vulnerability scanning combined with role-based access control and centralized reporting, SecureWebOps enables small organizations to improve their security posture using a single, shared platform without requiring enterprise-scale infrastructure or dedicated security teams.

Who Is Affected?

Our primary persona is Bob, Web Applications & DevOps Lead at a small organization. His team must deliver secure software without the tools or budget of a large enterprise. Bob represents small teams that need fast, accessible web vulnerability scanning today and secure add-ons like file encryption and phishing protection in the future.

Group Affected	**Problem They Face**	**How They Cope Without SecureWebOps**
Web Development Teams	Security scans interrupt workflows and slow down delivery when using traditional tools.	They skip scans or run them less often, increasing the risk of releasing vulnerable code.
Small IT/Security Teams	Lacks dedicated cybersecurity staff and cannot maintain complex enterprise tools.	They rely on free, fragmented tools or ignore security findings due to a lack of time and expertise.

Small Organizations (Schools, Nonprofits, Startups)	Cannot afford enterprise-level security software for scanning, encryption, and phishing defense.	They operate with security blind spots and rely on insecure ad hoc processes (emailing files, no scanning, and unencrypted sharing).

Empathy Map (Bob, the Web App Lead)

Empathy Map Category	**Insights about Bob**
Says	"Security slows down our delivery." • "We need something simple that doesn't require a security team."
Thinks	"We should be secure, but we can't afford complex tools." • "My team should not sacrifice speed or security."
Does	Leads deployment and development pipeline • Skips or postpones scans when deadlines approach • Uses free scanners occasionally
Feels	Stressed by deadlines • Frustrated by expensive tools • Worried about being blamed if a breach happens
Pains	Slow/full scans delay releases • Too many disconnected tools • Limited staff & time • Vulnerabilities go unnoticed
Gains	Wants fast, automated scans from the browser • Wants dashboards to track issues • Wants simple encryption workflows for shared files (future)

The empathy map helped our team understand Bob's frustrations and expectations as a real user who manages web applications with limited time and resources. Bob worries about slow security processes, is pressured by deadlines, and dislikes tools that interrupt development flow. He values tools that are fast, lightweight, and easy to understand without deep cybersecurity knowledge. These insights shape our product requirements: scans must run quickly, encryption must be automatic, and reports must be readable for non-experts. We prioritized an extension-based scanner to reduce complexity and a simple dashboard

with visual risk alerts, reflecting Bob's need for a tool that protects his work without slowing him down.

How SecureWebOps Addresses These Problems

SecureWebOps directly addresses the challenges faced by small teams by providing accessible, lightweight, and integrated security tools that do not slow down development. Design decisions for the platform are informed by both academic research and industry observations from Federal Resources Corporation (FRC), which emphasize usability and workflow integration over complex enterprise-scale tooling. The platform is intentionally built in phases to ensure that the most urgent and widely needed functionality of web vulnerability scanning is delivered first. Additional modules such as file encryption, phishing detection, and reporting dashboards enhance protection over time without requiring users to learn or maintain separate systems.

Fast, Integrated Web Scanning (MVP):

SecureWebOps introduces a browser-based vulnerability scanner that authorized users can run against web applications owned or managed by their organization without leaving their workflow. By launching scans directly from a browser extension, users can initiate vulnerability scans on their own organizational websites without installing standalone scanning tools or navigating complex security platforms.

Value: Website vulnerability scans become quick and accessible for small teams, reducing delays in identifying security issues during routine website maintenance and development.

Secure File Encryption for Sensitive Data (Future Module):

Many small organizations share configuration files, contracts, and documents that often include private information. SecureWebOps will provide automatic AES-based encryption for uploaded files, using centralized key management to ensure sensitive documents are stored and shared securely without requiring users to manage encryption keys directly.

Value: Reduces the risk of accidental data exposure by providing a consistent and secure method for handling sensitive files within the platform.

Phishing Detection + Email Security (Future Module):

Teachers, schools, and nonprofits increasingly face phishing attacks but lack dedicated cybersecurity staff. SecureWebOps will offer a phishing detection workflow that allows users to submit suspicious links or files associated with their organization and receive a basic risk assessment indicating potential phishing indicators.

Value: Enables non-technical users to perform an initial evaluation of suspicious content before taking further action or escalating to technical staff.

Unified Dashboard for All Security Activities:

Instead of juggling multiple tools, SecureWebOps consolidates scan results, alerts, and file security information into a single dashboard. Authorized users can view vulnerability scan results, encrypted file activity, and phishing assessment reports for their organization, with role-based access control used to restrict access based on user responsibilities.

Value: Small organizations can manage multiple security activities from a single interface, reducing reliance on multiple disconnected tools and minimizing operational overhead.

SecureWebOps simplifies common security tasks by providing browser-triggered vulnerability scanning and gradually expanding dashboard functionality to include encryption, phishing detection, and security reporting in later releases.

Expected Impact of this Computing/Engineering Solution

SecureWebOps is expected to improve cybersecurity awareness and baseline protection for small organizations that lack dedicated security staff or the resources required to deploy traditional enterprise security solutions. By simplifying vulnerability scanning through a browser extension and progressively integrating encryption, phishing detection, and reporting dashboards, SecureWebOps supports a secure-by-default approach by guiding users through predefined security workflows that do not require advanced cybersecurity expertise.

Technical Impact:

SecureWebOps reduces the complexity of adopting cybersecurity practices by replacing standalone vulnerability scanners with a browser-based scan request system that integrates directly into existing workflows, reflecting usability considerations commonly observed in industry environments. The platform supports secure configuration practices through role-based access control (RBAC), encrypted data handling, and controlled storage of security-related information. By embedding security actions into routine website management and development tasks, SecureWebOps enables developers and small IT teams to apply basic security workflows while minimizing disruption to productivity.

Economic Impact:

SecureWebOps lowers the cost of adopting foundational cybersecurity practices by reducing reliance on licensed tools, extensive training, and dedicated infrastructure commonly required by enterprise security solutions. Schools, nonprofits, startups, and small organizations can access core security capabilities at low cost, helping to reduce

disparities in access to cybersecurity tools between large organizations and smaller teams with limited resources.

Social Impact:

SecureWebOps encourages more proactive consideration of security risks during routine website maintenance and development activities. By increasing the accessibility of vulnerability scanning, secure file handling, and phishing awareness workflows, the platform supports safer digital practices for communities that manage sensitive data but may lack formal cybersecurity support.

Educational Impact:

SecureWebOps is designed to be usable by students and small teams, making it suitable for hands-on exposure to basic cybersecurity concepts and workflows. Through interaction with vulnerability scan results, encryption processes, and security reporting, users gain increased awareness of common vulnerabilities and security practices, supporting cybersecurity education and workforce development objectives.

In summary, SecureWebOps supports smaller organizations in identifying security issues, protecting sensitive data, and applying basic security practices that align with real-world operational constraints and resource limitations.

Stakeholders You Have Met

- Jeremy Young, Chief Executive Officer, Federal Resources Corporation

-Zack Maccario, Solutions Engineer, Federal Resources Corporation

- Alex Tayou, Cyber Security Specialist / Data Analyst, Federal Resources Corporation

At the current stage of the project, the SecureWebOps team has engaged with both real industry stakeholders and research-identified stakeholder groups to inform system requirements. One real stakeholder identified for this project is Federal Resources Corporation (FRC), a cybersecurity solutions integrator that works closely with government, defense, and enterprise customers to assess security needs and recommend appropriate technologies. While SecureWebOps is not intended to replicate enterprise-grade tools, interaction with an industry-focused stakeholder such as FRC provides insight

into common security challenges, expectations, and workflows observed across organizations with varying levels of security maturity.

In addition to this real stakeholder, requirements were refined using research-driven elicitation methods, including persona modeling, market analysis, cybersecurity frameworks, and industry reports. This approach is reliable because it combines practical industry perspectives with established security standards and documented real-world challenges faced by small organizations. As the project transitions into implementation and refinement, the team intends to further validate requirements through continued stakeholder feedback and iterative testing.

Stakeholder Analysis

Stakeholder Group	**Real-World Role**	**Why They Matter to SecureWebOps**	**How We Identified Their Needs (Requirement Elicitation Technique)**
Federal Resources Corporation (Industry Stakeholder)	Cybersecurity solutions integrator supporting organizations with diverse security needs	Provides real-world insight into common security gaps, usability challenges, and expectations across organizations without dedicated security teams	Industry discussions, observation of solution integration workflows, review of publicly available vendor documentation and best practices
Web Developers / DevOps Engineers	Build, deploy, and maintain web applications	Require low-friction vulnerability scanning that integrates into existing workflows	Persona modeling, OWASP Developer Reports, competitor feature analysis
Small Organizations (Schools, Nonprofits,	Organizations with limited budgets and small IT teams	Need affordable, integrated security tools rather than fragmented products	Market research, small-business cybersecurity reports (e.g., DBIR)

Startups)			
Security Analysts / IT Staff (Future User Group)	Investigate security issues and manage incidents	Need visibility into scan results, logs, and security findings	Review of SOC workflows, NIST CSF incident response guidelines
General Users Uploading Files or Reviewing Emails	Users who handle sensitive documents or flag suspicious content	Need simple, guided workflows for encryption and phishing review	Analysis of data protection standards, encryption best practices, privacy research

Requirement Elicitation Method Summary

Stakeholder needs for SecureWebOps were identified using a combination of industry-informed input and research-based requirement elicitation methods. An industry stakeholder, Federal Resources Corporation (FRC), provided perspective on common security challenges, usability constraints, and workflow expectations observed across organizations with varying levels of security maturity. This input helped validate assumptions related to tool complexity, access control requirements, and the need for integrated security workflows.

In addition to industry insight, persona development and empathy modeling were used to understand common frustrations experienced by developers and small IT teams, including slow vulnerability scans, fragmented security tools, and security processes that introduce workflow friction. Market research and competitor analysis further highlighted how existing tools often separate vulnerability scanning, encryption, phishing protection, and reporting into distinct products, reinforcing the need for consolidation.

Recognized cybersecurity frameworks and industry reports were also used to ground system requirements in established best practices. These sources included the OWASP Top 10 to identify common web vulnerabilities, the NIST Cybersecurity Framework to inform access control and logging requirements, and the Verizon Data Breach Investigations Report (DBIR) to assess attack trends impacting small organizations. Together, these methods

ensured that system requirements were informed by both real-world industry observations and established security standards.

Future Stakeholder Engagement Plan

As the project progresses into later development and evaluation phases, additional stakeholder engagement will be conducted to further validate and refine system requirements. Planned engagement includes interviews with small IT teams in local schools and nonprofit organizations, discussions with university IT and cybersecurity professionals, and feedback from small software startup developers. These interactions will be used to validate usability expectations, refine reporting workflows, and confirm role-based access assumptions identified during earlier stages of development.

Project Scope and Requirements

Project Scope

SecureWebOps is a unified cybersecurity platform designed for small organizations that lack the resources to deploy and manage multiple specialized security tools. The system consolidates core security capabilities—including website vulnerability scanning, phishing content evaluation, and secure file encryption—into a single, accessible platform. SecureWebOps emphasizes data security through encrypted storage, role-based access control, audit logging, and centralized reporting, allowing organizations to manage security tasks without enterprise-scale infrastructure or dedicated security teams.

The platform consists of two primary components: a browser extension used to initiate security actions and a web-based platform that hosts backend services, data storage, and reporting functionality.

The platform consists of two major components:

Component	**Purpose**
SecureWebOps Browser Extension	Allows authorized users to submit website URLs for vulnerability scanning directly from their browser.

SecureWebOps Web Platform	Hosts backend services including vulnerability scan processing, secure file encryption, phishing analysis, data storage, and security dashboards.

The scope includes:

- Secure File Encryption and Protected Storage

Encryption of uploaded files (e.g., PDFs) using centralized key management, along with controlled storage and access restrictions.

- Phishing Content Evaluation (Future Module)

Analysis of user-submitted links or files to identify potential phishing indicators; direct email inbox integration is out of scope.

- Automated Website Vulnerability Scanning (MVP)

Scans are initiated through browser extension submission of organization-owned or managed website URLs; future releases may support direct submission through the web platform.

- Security Dashboard and Reporting

Centralized dashboard displaying scan results, file activity, and phishing assessments.

- Role-Based Access Control (RBAC)

Support for multiple users per organization with defined roles and access permissions.

- Logging and Audit Trail

Logging of security actions and events for visibility and basic auditing.

- Centralized Key Management

Key management to support encrypted file handling and secure data storage.

Functional Requirements (System Shall …)

Functional Requirement	**Description**

Multi-Tenant Organization Support	The system shall support multiple organizations within a single platform instance, ensuring logical separation of users, data, scan results, and encrypted files between organizations.
Website Scan Submission via Browser Extension (MVP)	The system shall allow authorized users to submit organization-owned or managed website URLs through a browser extension to initiate vulnerability scans.
Backend Scan Processing	The system shall queue and execute vulnerability scans on submitted URLs using an approved scanning engine (e.g., OWASP ZAP) and associate results with the submitting organization.
Secure File Upload and Encryption (Future Module)	The system shall allow authorized users to upload files (e.g., PDFs) that are encrypted prior to storage using centralized key management.
Encrypted Document Retrieval and Decryption	The system shall allow only authorized users within the same organization to retrieve and decrypt encrypted documents.
Phishing Content Evaluation (Future Module)	The system shall allow users to submit suspicious links or files associated with their organization for phishing evaluation and return a basic risk assessment; direct email inbox integration is out of scope.
Dashboard Viewing and Reporting	The system shall display vulnerability scan results, security alerts, file activity, and phishing assessments in a centralized dashboard scoped to the user's organization.
Role-Based Access Control (RBAC)	The system shall enforce role-based permissions (e.g., Administrator, Analyst, Basic User) to restrict access to security actions and data based on user responsibilities.
Audit Logging	The system shall log security-relevant events, including scan submissions, file uploads, encryption and decryption attempts, and authentication events.
Centralized Key Management	The system shall manage encryption keys using a centralized key management mechanism to support secure file encryption and

	storage.
Secure Communication	The system shall enforce HTTPS-only communication for all client-to-server and server-to-server interactions.

Brief Use Case Descriptions

Use Case Name	**Actor(s)**	**Triggering Event**	**Brief Description**
Submit Website for Vulnerability Scan	Authorized User	The user submits URL via browser extension or dashboard	The system initiates a vulnerability scan for an organization-owned website and stores results.
View Security Dashboard	User, Administrator	User accesses the dashboard.	The system displays vulnerability scan results, file activity, and phishing assessments scoped to the user's organization.
Upload and Encrypt File	Authorized User	User uploads file	The system encrypts the file and stores it securely for authorized organizational access.
Submit Content for Phishing Evaluation	Authorized User	The user submits suspicious link or file	The system evaluates content for phishing indicators and returns a risk assessment.
Manage User Accounts and Roles	Administrator	Admin updates users or roles	The system creates, updates, or restricts user accounts based on role.
Perform Quick Scan via Browser Extension	Authorized User	User clicks extension icon	The system submits the active site URL for scanning and provides status feedback.

View Security Activity History	Authorized User	The user selects history view	The system retrieves and displays past scans, file actions, and phishing evaluations.
Receive Security Alerts	System	Scan or analysis completes	The system notifies users of scan results or detected risks.

The MOST Complicated Use Case: Secure File Encryption & Retrieval

Use case	**Secure File Encryption & Retrieval**
Scenario	The user uploads a file containing sensitive information to SecureWebOps for secure storage and later retrieval by authorized organization members.
Triggering event	The user submits a file upload request through the SecureWebOps platform.

Brief Description	The system validates the uploaded file and the user's authorization, encrypts the file using centralized key management, and stores the encrypted file in organization-scoped storage. Authorized users can later retrieve and decrypt the file. All actions are logged for auditing and compliance purposes.
Actors	User (Admin or Standard User)
Related use cases	View Unified Security Dashboard & Reports, Manage User Roles, Receive Security Alerts
Stakeholder	Small organizations that need a simple, consistent way to protect sensitive documents without managing encryption tools or keys.
Preconditions	• The user must be authenticated and associated with an organization of tenants. • The user must have permission to upload or retrieve files.

	• Key management services must be available.	
Postconditions	• The file is stored in encrypted form and associated with the organization. • Upload, encryption, and retrieval actions are recorded in the audit log. • Authorized users can securely retrieve the file.	
Flow of activities	Actor	System

1. The user submits a file for secure storage.	1.1 The system validates file type, size, and user permissions.
	2.1 The system requests an encryption key from the key management service.
2. The user confirms the upload request.	
	2.2 The system encrypts the file using centralized key management.
	3.1 The system stores the encrypted file in organization-scoped storage.
3. The user waits for confirmation of secure storage.	
	3.2 The system logs the upload and encryption event in the audit log.
4. The user later requests access to the stored file.	4.1 The system verifies retrieval permissions using role-based access control.
	4.2 The system decrypts the file using centralized key management.
	4.3 The system delivers the file securely and logs the retrieval event.

Exception Conditions	• File type or size violates platform restrictions. • The user lacks permission to upload or retrieve files (RBAC restriction at steps 1.1 or 4.1). • Encryption or decryption fails due to key management service unavailability. • The user attempts to access a file belonging to a different organization.

2nd MOST Complicated Use Case: View Unified Security Dashboard & Reports

Use case	**View Unified Security Dashboard & Reports**
Scenario	The user accesses the SecureWebOps dashboard to review security-related information for their organization, including scan results, encrypted file activity, and phishing assessments.

Triggering event	The user navigates to the SecureWebOps dashboard after authenticating.
Brief Description	The system retrieves organization-scoped security data, including vulnerability scan summaries, encrypted file activity, phishing assessment results, and audit logs. The system filters displayed information based on the user's role and permissions and presents the data in a centralized dashboard for monitoring and decision-making.
Actors	User (Admin or Standard User)
Related use cases	Organization-Managed Website Vulnerability Scan, Secure Document Encryption and Retrieval, Receive Security Alerts, Log User Actions

Stakeholder	Small organization administrators and IT generalists who need centralized visibility into security activities without managing multiple tools.	
Preconditions	• The user must be authenticated. • The user must belong to an organization tenant. • The user must have permission to view dashboard data.	
Postconditions	• Security data is displayed according to the user's role and organization. • Dashboard access is recorded in the audit log.	
Flow of activities	Actor	System

1. The user accesses the SecureWebOps dashboard.	1.1 The system authenticates the session and identifies the user's organization.
2. The user requests to view security information.	2.1 The system retrieves security data scoped to the organization.
3. The user selects a report or summary view.	2.2 The system applies role-based access control to filter visible data.
4. The user reviews security findings and activity history.	3.1 The system retrieves the requested report data.
	4.1 The system displays summarized metrics and detailed reports.
	4.2 The system logs dashboard access for auditing purposes.

Exception Conditions	• The user session is expired or invalid. • The user lacks permission to view the requested data (RBAC restriction). • Required security data is temporarily unavailable due to backend service issues.

Complete Non-functional Requirements

These describe *how well* the system must work.

Category	**Requirement**	**Measurement**
Usability	Quick Scan Accessibility	At least 90% of first-time users must be able to initiate a basic website scan in three clicks or fewer from the dashboard or browser extension. This will be validated through peer usability testing conducted during course demonstrations and internal team testing.

Efficiency	Scan Response Speed	Upon scan submission, the system must acknowledge the request and display a “scan started” or “queued” status within 5 seconds. For low-complexity websites, scan results should be available on the dashboard within 3 minutes under normal operating conditions.
Security/Privacy	Secure Key and Scan Data Protection	All encryption keys must be stored and managed outside the application using an external key management service (KMS). Keys must never be stored or transmitted in plain text. Compliance will be validated through configuration review, encryption verification, and audit logging.
Reliability/Availability	System Availability	The SecureWebOps backend services must maintain at least 99% uptime during active project demonstration periods, excluding scheduled maintenance, as measured through application logs and basic monitoring alerts.
Maintainability	Updatable Scan Rules	The system must support updating vulnerability scanning rules and configurations (e.g., OWASP ZAP rules) without requiring full system redeployment. Updates should complete within 10 minutes during maintenance windows.

Standards, Cybersecurity Frameworks, Benchmarks and Constraints

Key Standards and Security Frameworks

We base our work on established industry guidelines to ensure a high level of security.

Standard / Framework	**What is it? (Summary)**	**Official URL**	**How We Apply It**

OWASP Top Ten	A widely adopted industry list identifying the ten most critical security risks affecting modern web applications.	https://owasp.org/www-project-top-ten/	SecureWebOps aligns its website vulnerability scanning functionality with OWASP Top Ten categories. The backend scanning engine (OWASP ZAP) is configured to detect common OWASP risks such as Cross-Site Scripting (XSS), SQL Injection, insecure HTTP headers, and security misconfigurations. Scan findings are mapped to OWASP categories, assigned severity levels, and displayed in the dashboard so users can prioritize remediation based on recognized industry risk classifications.
NIST Cybersecurity Framework (CSF)	A structured framework for managing cybersecurity risk using five core functions: Identify, Protect, Detect, Respond, and Recover.	https://www.nist.gov/cyberframework	SecureWebOps applies NIST CSF concepts across multiple system components. Identify is supported through organization-scoped asset and scan tracking. Protect is implemented using role-based access control (RBAC) and encrypted storage. Detect is addressed through vulnerability scanning, phishing analysis, and audit logging. Respond is supported by dashboard alerts, scan reports, and risk summaries

			that guide corrective action. Recovery concepts are acknowledged through secure data storage and historical reporting for future review.
AES-256 Encryption Standard	A globally accepted symmetric encryption standard used to protect sensitive data at rest.	https://csrc.nist.gov/projects/block-cipher-techniques	SecureWebOps uses AES-256 encryption for protecting sensitive files (such as uploaded PDFs) and stored security metadata. Encryption keys are generated and managed through an external Key Management Service (KMS), ensuring keys are never hard-coded or stored in plain text. Encryption and decryption operations occur server-side, and all related actions are logged to support auditing and compliance verification.

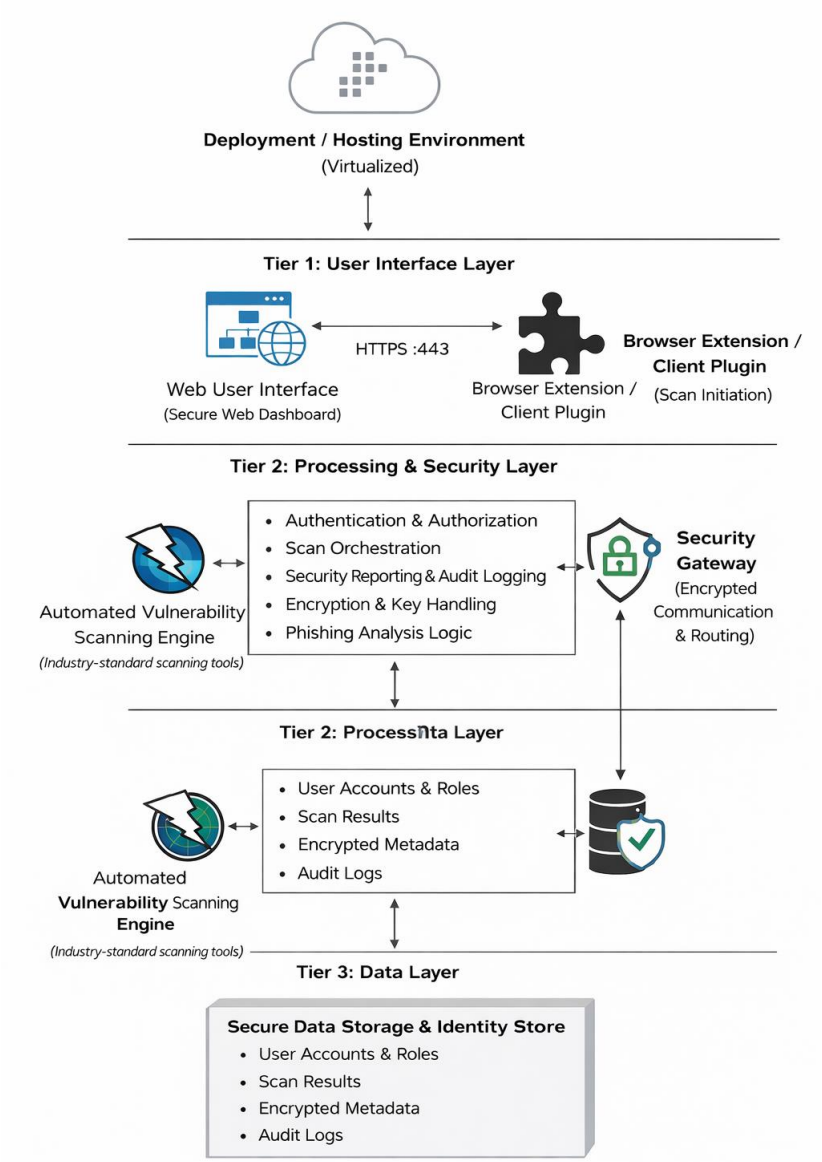
Project Constraints

Constraint	Why is it a problem?	How We Handle It
Project Timeline	We only have so much time, which limits how many features we can fully build.	We are using an Agile approach and focusing strictly on delivering a working MVP (Minimum Viable Product) prototype of the core features.
Budget Limitations for Small Orgs	Users cannot afford enterprise tools like Burp Suite or Prisma.	We use free/open-source tools (OWASP ZAP, Supabase free tier, Caddy).

Limited Technical Security Skills of Users	Users cannot manage complex configurations.	We provide a simple UI + browser extension + automated scanning.
Browser Extension Security Restrictions	Browser APIs cannot store secrets safely.	No secret keys stored in extension; all authentication handled by backend via HTTPS.

System Architecture

High-Level Architectural Diagram



User Interface (UI) Layer:

The user interface layer consists of a secure web-based dashboard and a client-side browser extension or plugin. The browser extension enables authorized users to submit organization-owned website URLs for vulnerability scanning directly from their workflow. The web dashboard provides centralized access to vulnerability scan reports, security findings, and user account management. Future system phases will extend this interface to support secure file encryption workflows and phishing detection features. This layer focuses on usability, accessibility, and secure interaction with backend services while remaining independent of specific implementation technologies.

Security & Routing Layer:

The Security and Routing layer functions as a secure gateway between client interfaces and internal system services. It enforces encrypted communication for all inbound and outbound traffic, manages secure request routing, and ensures that only authenticated and authorized requests are forwarded to backend components. This layer provides foundational network-level security controls and abstracts transport security mechanisms from application logic.

Processing Layer:

The processing layer contains the core application logic of the system. It is responsible for enforcing authentication and authorization policies, orchestrating vulnerability scanning operations, generating security reports, and logging user and system activity for audit purposes. This layer validates all client requests and coordinates interactions between the user interface, scanning services, and data storage components. Additional security-focused logic, such as file encryption workflows and phishing analysis, is integrated into this layer to ensure sensitive operations are handled centrally and securely.

Data Layer:

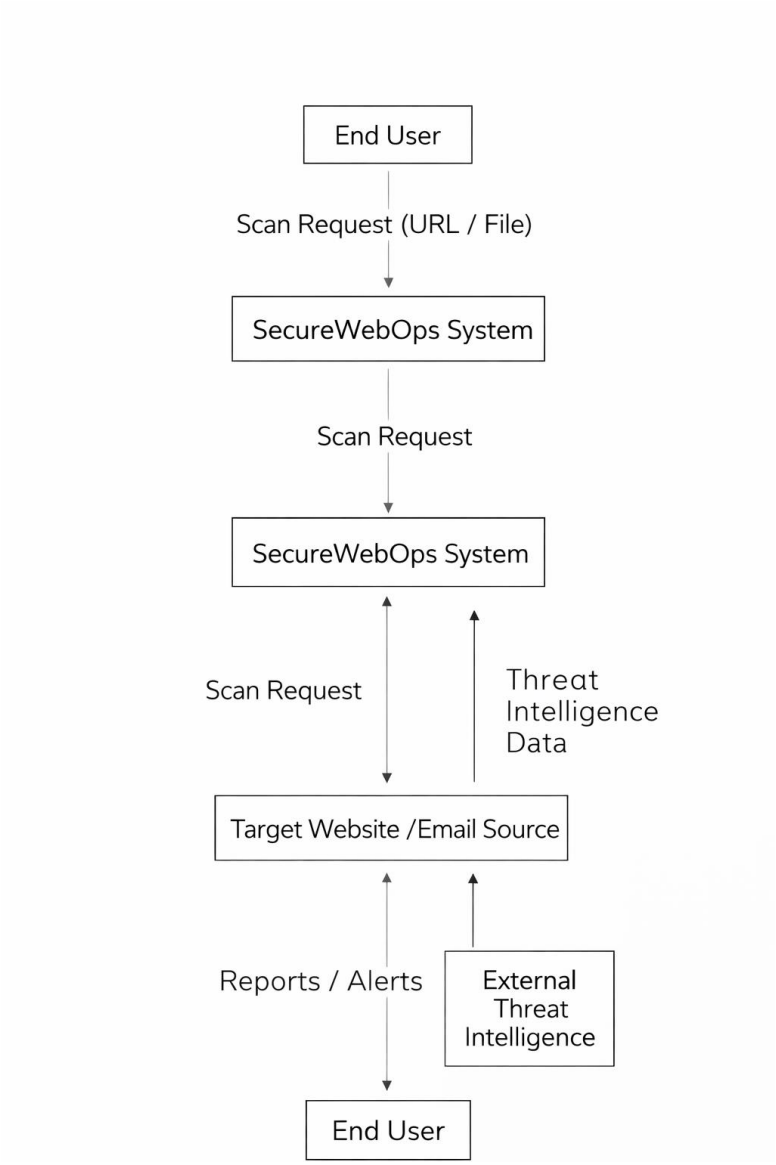
The data ility scan results, encrypted metadata, and audit logs. It enforces strict access control policies to maintain isolation between organizational tenants and users. A centralized key management mechanism governs cryptographic keys used for protecting sensitive files and data, ensuring keys are never exposed to client devices or browser extensions. This layer is designed to support secure storage, compliance, and long-term data integrity independent of specific database or key management technologies.

Internal Vulnerability Scanning Engine:

The system integrates an internal automated vulnerability scanning engine responsible for executing security assessments against authorized, organization-managed web applications. Scan requests are scheduled and initiated by the processing layer, and structured scan results are returned for storage and reporting. The scanning engine follows industry-recognized vulnerability assessment practices and is designed to be modular, allowing scanning technologies to be updated or replaced without impacting the overall system architecture.

Context Diagram

This context diagram represents SecureWebOps as a single system and illustrates its interactions with external entities at a high level. The system receives security-related requests—such as website scan submissions or content analysis—from end users through a browser-based interface. SecureWebOps interacts with external target systems, such as organization-managed websites or email sources, to perform security assessments. The system may also reference external threat intelligence sources to enhance detection accuracy. Scan results, reports, and alerts are returned to the end user for review. Internal processing details and implementation technologies are intentionally abstracted in this diagram.

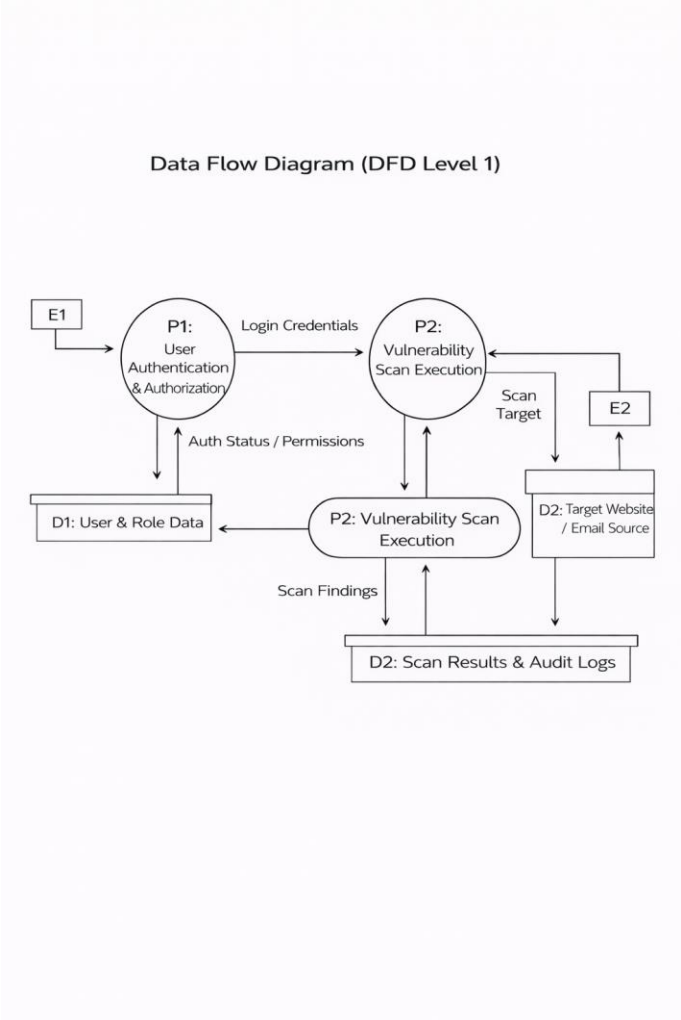


This Level 0 context diagram intentionally abstracts internal system components and technologies to focus on external entities and high-level data flows, following standard context diagram conventions.

- End User: Submits security-related requests (e.g., website scan requests) and receives reports or alerts from the system.

- Target websites/emails are evaluated for potential vulnerabilities or phishing indicators.

- External Threat Intelligence Sources: External data providers that supply up-to-date information on known vulnerabilities, attack patterns, or malicious indicators used to enrich security analysis.



Data Flow Diagram (DFD Level 1) – Process & Data Store Descriptions

- P1: User Authentication & Authorization

Validates user credentials submitted by the end user and determines access permissions based on assigned roles. This process retrieves and updates user and role information from the User & Role Data Store and returns authentication status and permissions to the user.

- P2: Vulnerability Scan Execution

Receives authorized scan requests from authenticated users and performs security assessments against approved target websites or email sources. This process sends scan requests to external target systems, collects scan findings, and forwards the results for reporting without permanently storing data locally.

- P3: Reporting & Audit Logging

Receives scan findings from the vulnerability scanning process, stores scan results and security events in the Scan Results & Audit Logs data store, and retrieves historical data to generate reports or alerts for the end user.

-D1: User & Role Data

Stores user account information and role assignments used to enforce authentication and authorization decisions.

- D2: Scan Results & Audit Logs

Stores vulnerability scan results and system activity logs to support reporting, auditing, and accountability.

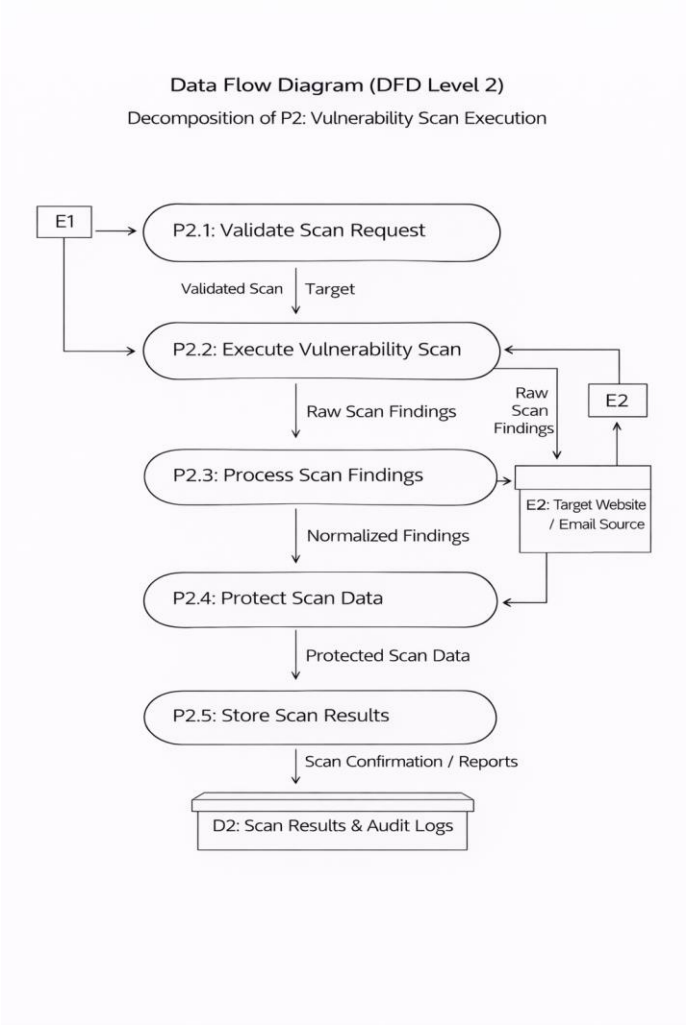
- E1: End User

Initiates authentication requests, submits vulnerability scan requests, and receives scan reports or alerts from the system.

- E2: Target Website / Email Source

Represents external systems owned or managed by the organization that are evaluated for vulnerabilities or phishing indicators during security assessments.

This DFD Level 1 follows standard data flow diagram notation by representing logical processes, data stores, external entities, and data flows while abstracting implementation technologies.



Data Flow Diagram (DFD Level 2) – Decomposition of P2: Vulnerability Scan Execution

- -P2.1: Validate Scan Request

Receives a scan request from the end user and verifies that the target URL or content source is properly formatted and authorized for scanning before initiating the security assessment.

- -P2.2: Execute Vulnerability Scan

Performs a security assessment against the approved target website or email source and collects raw vulnerability findings generated during the scan.

-P2.3: Process Scan Findings

Parses and normalizes raw scan output into a structured format suitable for reporting, storage, and further analysis.

- -P2.4: Protect Scan Data

Applies data protection mechanisms to processed scan findings to preserve confidentiality and integrity prior to storage.

-P2.5: Store Scan Results

Stores protected scan results and associated audit information in the Scan Results & Audit Logs data store and generates confirmation that results are available for reporting.

- -D2: Scan Results & Audit Logs

Stores vulnerability scan results and security-related event logs to support reporting, auditing, and historical analysis.

- -E1: End User

Submits vulnerability scan requests and receives scan confirmations or reports.

- E2: Target Website / Email Source

The external system was evaluated during the vulnerability scanning process.

-E2: Target Website / Email Source

External system evaluated during the vulnerability scanning process.

This DFD Level 2 decomposes the Vulnerability Scan Execution process identified in DFD Level 1 and follows standard data flow diagram notation by abstracting implementation technologies and focusing on logical data transformations.

System sequence diagrams, the process, and the most complicated individual process in the project.

This section provides formal interaction models describing (1) the overall system process and (2) the

most complicated individual process in SecureWebOps. These diagrams complement the data flow.

Diagrams illustrate runtime behavior between actors and system components.

1. Overall System Process – Website Vulnerability Scan

1. The end user submits a URL for vulnerability scanning through the user interface.

2. The authentication & authorization service validates the session token and role-based permissions.

3. The User & Role Data Store is queried to verify organizational scope.

4. If authorized, the request is forwarded to the Vulnerability Scan Execution process.

5. Vulnerability Scanning: The engine performs a security assessment against an approved target.

6. Raw findings are returned and normalized with severity classifications.

7. Reporting & Audit Logging stores results in Scan Results & Audit Logs Data Store.

8. The dashboard displays scan results and a risk summary to the end user.

2. Most Complicated Process – Secure File Encryption & Retrieval

Part A – File Upload and Encryption

1. The authorized user uploads the file through the web dashboard.

2. Authentication & Authorization validates tenant scope and upload permissions.

3. The backend requests an encryption key from the centralized Key Management Service (KMS).

4. The file is encrypted using AES-based encryption server-side.

5. The encrypted files are stored in secure storage.

6. Metadata and key references are stored in audit logs & metadata stores.

7. Upload and encryption events are logged.

Part B – File Retrieval and Decryption

1. The authorized user requests encrypted file retrieval.

2. Authentication & Authorization verifies access rights within the tenant.

3. The encrypted files and metadata are retrieved from storage.

4. Backend requests key unwrapping from KMS.

5. The file is decrypted server-side.

6. Retrieval and decryption events are logged.

7. The file is returned securely over HTTPS to the user.

Why This Is the Most Complicated Process

This is the most complex transaction, as it requires a strict implementation of role-based access (RBAC), multi-tenancy, centralized key management, cryptographic operations, secure storage, and tamper-evident audit logging. All these transactions must occur in a specific order, while ensuring that the encryption key(s) cannot be found on the client side.

Database/Data Model Design

The following Entity-Relationship Diagram (ERD) models how SecureWebOps securely

manages users, scans, encryption keys, logs, and encrypted data. The design enforces zero-trust separation of duties by isolating encryption keys from stored data and linking every user

action to an audit record.

ERD Overview

- Users & Roles enforce role-based access control (RBAC).
- ScanJobs & ScanFindings track scan requests and the vulnerability results produced by OWASP ZAP.
- EncryptedFiles & KMS_Keys ensure that sensitive data (like PDFs or future encrypted exports) never leave the backend unprotected.
- Audit logs capture non-repudiation logs for every action, preventing tampering or deniable misuse.

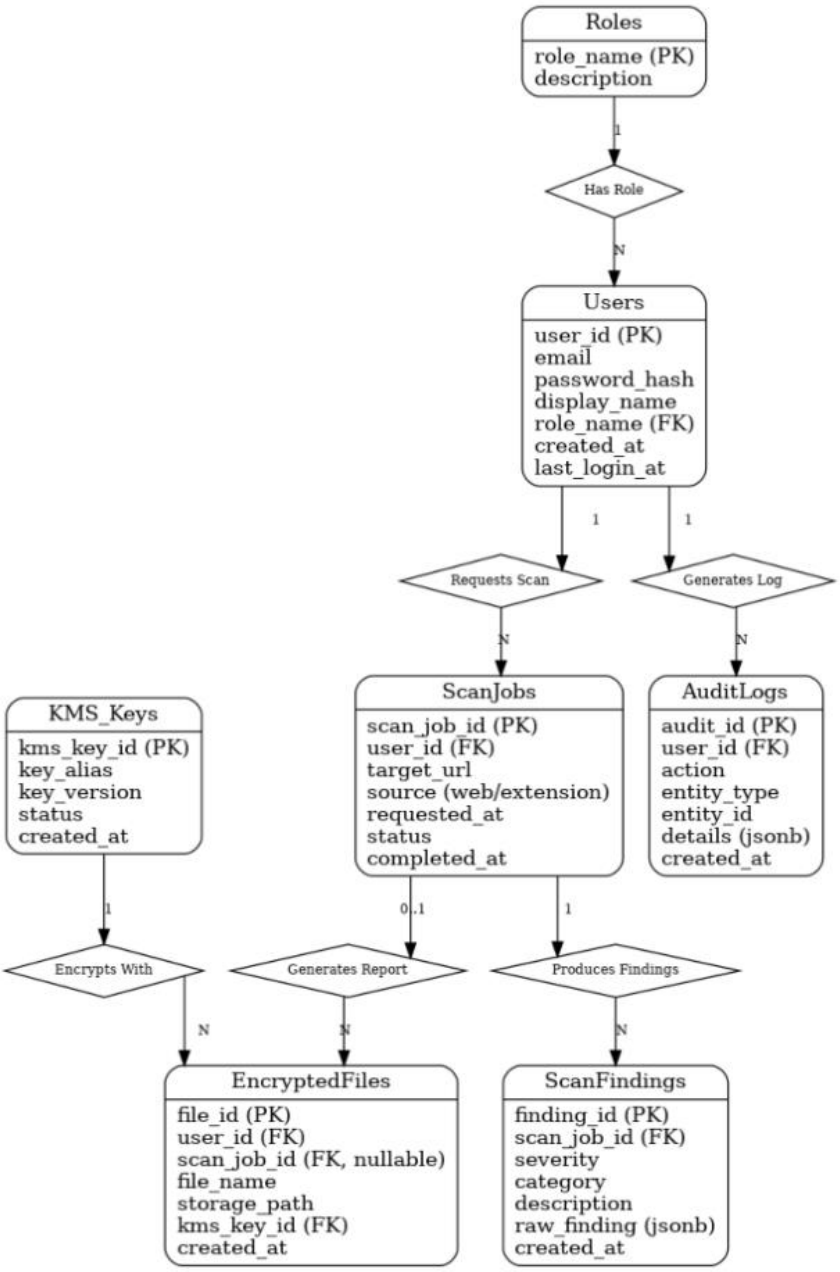
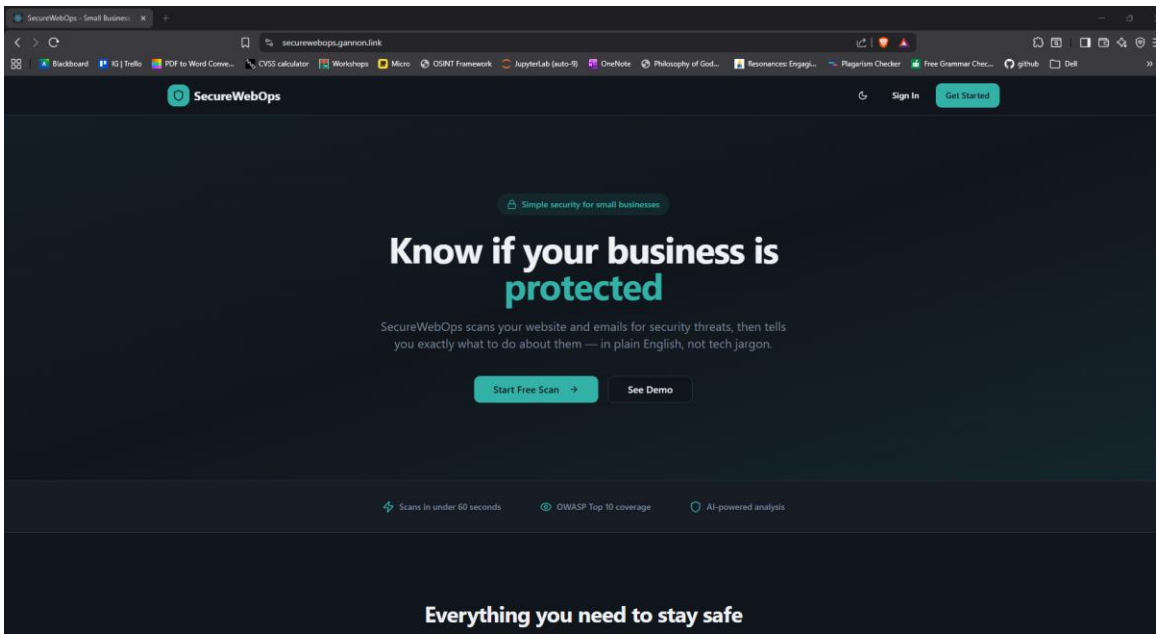


Table	Purpose	Key Security Feature
Roles	Defines RBAC roles (Admin, Standard User)	Prevents privilege escalation
Users	Stores hashed credentials & role assignment	No plaintext passwords; least privilege enforced

ScanJobs	Tracks each scan request made through UI/extension	Ties scans to authenticated users to prevent abuse
ScanFindings	Stores vulnerabilities and severity data	Findings are saved after KMS encryption
EncryptedFiles	Stores encrypted files (PDFs, reports)	Uses backend-only KMS keys (not browser keys)
KMS_Keys	Backend-managed encryption keys	Keys never leave secure environment
AuditLogs	Records user actions for non-repudiation	Enables tamper-evident logging

User Interface / Experience Design



Everything you need to stay safe

No security expertise required. We translate complex threats into simple action items.

Website Security Scans

AI-powered vulnerability detection that checks your website against OWASP Top 10 threats in plain language.

Phishing Detection

Paste any suspicious email or link and get an instant risk assessment with clear explanations.

Security Score

See your overall security health at a glance with a simple 0-100 score and actionable recommendations.

Team Management

Invite your team, assign roles, and keep everyone informed about your security status.

Built for busy business owners

You didn't start a business to become a cybersecurity expert. SecureWebOps handles the technical stuff so you can focus on what matters.

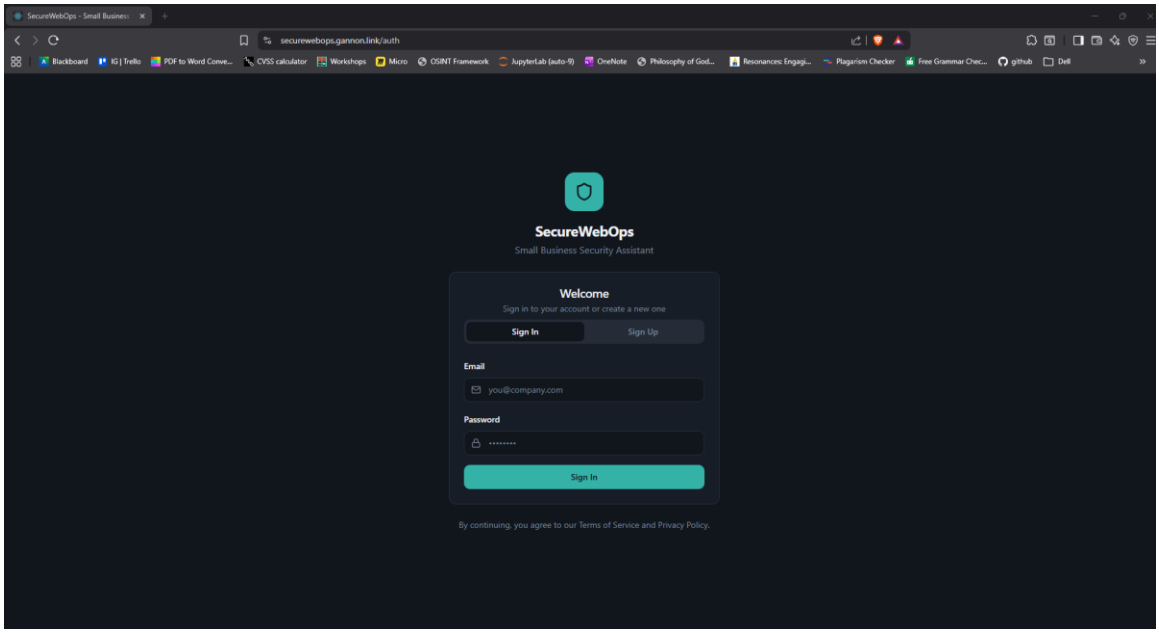
- No technical expertise required
- Results in plain English, not jargon
- Actionable recommendations you can follow
- Affordable for small businesses
- Email alerts for critical issues
- Track your security over time

Find out if your business is protected

Get your first security scan free. No credit card required. Results in under a minute.

Get Started Free →

SecureWebOps



The SecureWebOps user interface is designed to provide small business users with a clear, intuitive, and non-technical entry point into website and email security analysis. The primary goal of the interface is to translate complex security findings into actionable insights that can be easily understood and acted upon without requiring prior cybersecurity expertise.

Landing Page

The landing page serves as the main entry point to SecureWebOps. It introduces the platform's purpose—helping small businesses understand whether their digital assets are protected—and highlights core capabilities such as website security scanning, phishing detection, security scoring, and team management. Clear call-to-action elements allow users to initiate a free scan, view a demo, or proceed to account registration or sign-in. The landing page emphasizes simplicity, speed, and clarity, reinforcing that results are delivered in plain language rather than technical jargon.

Authentication and Access

From the landing page, users can access the authentication interface to securely sign in or create an account. Authentication acts as a gateway to personalized system features and

ensures that users only access data and reports associated with their organization and assigned role.

Dashboard

After authentication, users are presented with a centralized dashboard that functions as the main control center of the system. The dashboard provides a high-level overview of the organization's security posture, including a security score and indicators related to common risk areas. From this interface, users can initiate new website scans, review recent activity, and access historical results. The dashboard is designed to surface critical information first while allowing deeper exploration of findings as needed.

Scanning and Analysis

The interface allows users to initiate security scans by submitting website URLs or other supported inputs. Scan initiation is streamlined to reduce friction and guide users through the process with minimal required input. Progress and completion feedback is provided to ensure transparency throughout the scanning process.

Reports and Results

Scan results are presented in a reports view that lists previous scans along with relevant metadata such as the target analyzed, security score, scan date, and a summary of findings. Reports are designed to balance technical accuracy with readability, using clear language and visual indicators to communicate risk levels and recommended actions.

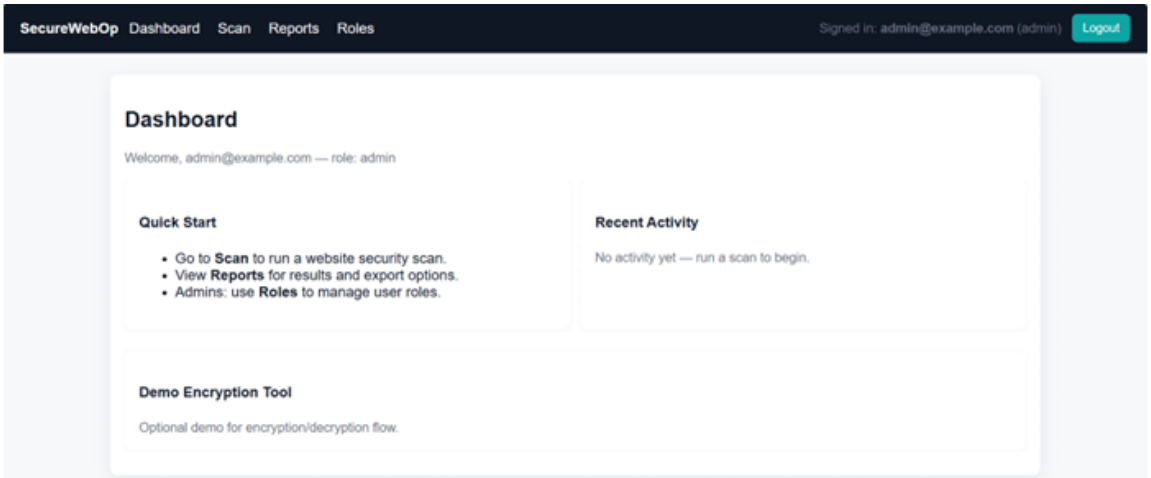
User and Team Management

The interface supports basic team and role management features, allowing organizations to control access to scans, reports, and administrative functions. Role-based access ensures that users only see information appropriate to their responsibilities while enabling collaboration within an organization.

Overall UX Considerations

The SecureWebOps UI prioritizes clarity, consistency, and accessibility. Visual hierarchy, simple navigation, and concise explanations are used throughout the interface to reduce cognitive load and make security information approachable. The design supports incremental feature expansion while maintaining a consistent user experience across system components.

The user interface is designed to abstract technical complexity while providing users with clear, actionable security insights through a consistent and intuitive workflow.



Website Scanner

Enter a URL to scan for common issues

Run Website Scan

https://campsite.bio/gufinegan

Start Scan | Clear

Scan Result

URL
https://campsite.bio/gufinegan
Scanned: 10/11/2025, 23:41:57

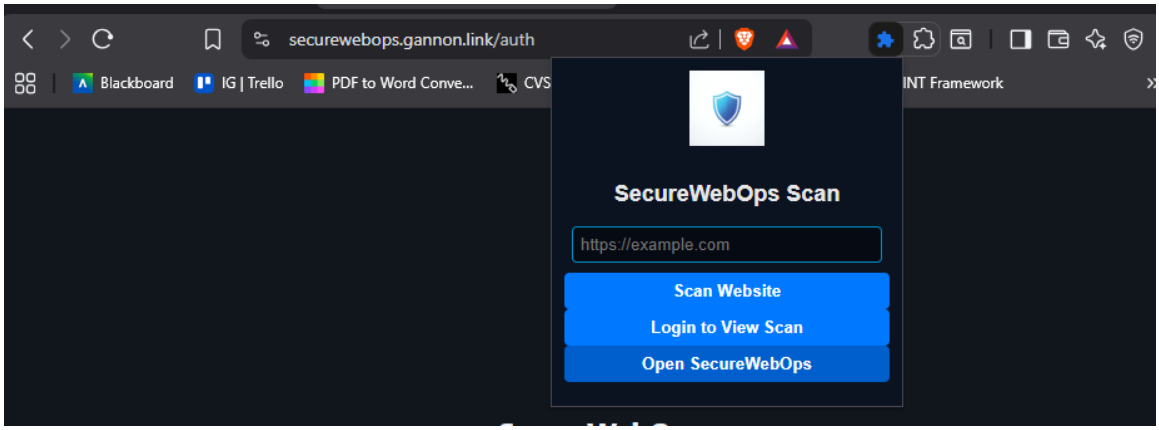
Security Score
84%

Findings

Reports

Saved scan reports

URL	Score	Scanned
https://campsite.bio/gufinegan	84%	10/11/202 23:41:57
https://www.gannon.edu/academic-offerings/engineering-and-business/institute-for-health-and-cyber-knowledge/	76%	06/11/202 12:49:02
https://jenkins.ihk.network/	79%	06/11/202 01:41:45
https://owl.purdue.edu/owl/research_and_citation/mla_style/mla_formatting_and_style_guide/mla_works_cited_electronic_sources.html	78%	06/11/202 01:39:03
https://knightsgannon-my.sharepoint.com/:p:/r/personal/moore102_gannon_edu/_layouts/15/Doc.aspx?sourcedoc=%7B11BA2297-AA0F-406A-9AD6-D020F44D06E1%7D&file=SecureWebOps%20Demo.pptx&fromShare=true&action=edit&mobileredirect=true	93%	05/11/202 20:57:46
https://gannon.blackboard.com/ultra/institution-page	77%	05/11/202 19:06:57
https://jenkins.ihk.network/	75%	05/11/202 15:33:37



The SecureWebOps browser extension provides a lightweight client-side interface that allows users to initiate vulnerability scans directly from their browser. The extension enables users to submit a target website URL for analysis with minimal interaction, reducing friction in the scanning workflow and supporting rapid security assessment.

Upon submission, the extension forwards the scan request to the SecureWebOps system through a secure communication channel. The extension informs users that scan processing and result generation occur within the main SecureWebOps platform and that detailed findings will be available through the authenticated web dashboard. This design ensures that sensitive data and analysis results are not stored locally within the browser environment.

The browser extension serves as an entry point for scan initiation and complements the web-based dashboard by enabling users to trigger scans from their existing workflow. Functionally, this interface supports the Vulnerability Scan Execution process (P2) defined in the Data Flow Diagram (DFD Level 1) and aligns with the Client/User Interface layer of the system architecture by providing a secure, minimal, and user-friendly mechanism for scan submission.

The browser extension is intentionally limited to scan initiation and user notification, with all processing, storage, and reporting handled by centralized system components.

Accessing the Extension:
The source code for the SecureWebOps Chrome extension is publicly available in our team's GitHub repository:
🔗 https://github.com/Blossom-Anolue/SecureWebOpsApp.git

Since the repository is public, the extension can be downloaded or cloned directly for installation and future development. No sensitive keys or backend credentials are stored inside the extension to ensure secure usage in an open-source environment.

Installation Steps:

1. Open Google Chrome and navigate to:
   chrome://extensions/

2. Enable Developer mode (toggle in the top right).

3. Click Load unpacked.

4. Select the extension/ folder from the cloned GitHub repository.

Once loaded, the Shield icon will appear in Chrome. Users can enter a URL to submit a scan request.

Current Functionality

At this development stage, the extension:

- Accepts a user-entered target URL
- Validates that the input is not empty
- Displays confirmation messages
- Informs users that results will appear in the SecureWebOps dashboard

This version acts as a secure frontend trigger, without storing local data or executing scans directly, supporting the project's zero-trust approach.

Technology Stack and Design Justification

Technology	Category	Role in SecureWebOps	Design Justification
JavaScript / TypeScript	Programming Language	Core language for browser extension, web dashboard, and backend services	Chosen for strong compatibility across frontend and backend components, high developer productivity, and team familiarity. TypeScript improves maintainability and reduces runtime errors.
HTML / CSS	Web Technologies	User interface structure and styling for the secure web dashboard	Universally supported by browsers and well-suited for building accessible, responsive interfaces with low maintenance overhead.
Chrome / Chromium Extension APIs (Manifest V3)	Platform / APIs	Enables browser extension functionality such as scan initiation and permission handling	Aligns with modern browser security requirements, supports secure permission models, and ensures compatibility with current extension standards.
Node.js	Application Runtime	Backend processing services and scan orchestration logic	Well-suited for I/O-heavy API workflows, supports asynchronous processing, and aligns with the team's skillset. Offers strong ecosystem support and maintainability.
Caddy	Reverse Proxy / Security Gateway	Enforces HTTPS, handles TLS, and securely routes client requests to backend services	Provides automatic TLS management, strong secure defaults, low configuration complexity, and reduces operational overhead while improving security.
PostgreSQL (via Supabase)	Database	Stores user accounts, roles, scan results, encrypted metadata, and audit logs	Reliable relational database with strong query capabilities for reporting and auditing. Cost-effective, scalable, and widely supported.
Supabase Authentication + RBAC + Row-Level Security	Identity & Access Control	Authentication, authorization, and enforcement of role-based access controls	Reduces custom security code, enforces least-privilege access through RLS, and integrates directly with the database for secure data isolation.

Web Crypto API	Cryptography API	Standards-based cryptographic operations for client-side workflows	Provides native browser cryptography without exposing raw keys, supports secure algorithms, and avoids insecure custom implementations.
OWASP ZAP	Vulnerability Scanning Tool	Automated vulnerability scanning engine used within the scan pipeline	Industry-recognized open-source scanner aligned with OWASP Top 10 vulnerabilities, cost-effective, and well-documented for maintainable integration.
JSON	Data Format	Structured data exchange for scan requests, results, and API communication	Lightweight, human-readable, and universally supported format that simplifies integration, logging, and storage of scan findings.

Why this stack fits the project

This stack was selected to align with SecureWebOps requirements for a browser-based workflow, secure multi-user access, repeatable scanning, and centralized reporting/logging:

-Compatibility with requirements: JavaScript/TypeScript across extension, dashboard, and backend reduces integration friction. Supabase + Postgres supports structured storage for scan results and audit logs, while OWASP ZAP supports vulnerability scanning aligned to OWASP-focused requirements.

-Performance: Node.js is well-suited for API orchestration and I/O-heavy workflows. Postgres supports efficient querying for reports and historical analysis. Web Crypto provides optimized browser-native crypto operations where applicable.

-Team skillset: The stack uses common web technologies (JS/TS, Node.js, Postgres) to maximize development velocity and reduce onboarding time.

-Cost and maintainability: Supabase reduces the need to build authentication, RBAC, and database tooling from scratch. Caddy reduces operational overhead by automating TLS and providing secure defaults. OWASP ZAP is open-source and commonly used.

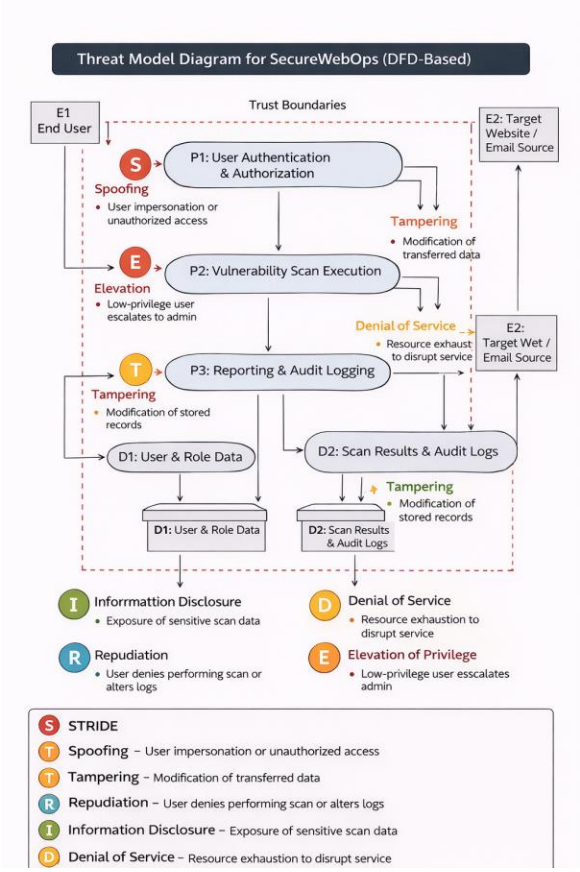
-Security and compliance: Caddy enforces HTTPS, Supabase RLS supports least privilege and data isolation, and Web Crypto avoids insecure custom cryptography. The overall stack supports secure-by-default implementation and aligns with common security best practices (RBAC, audit logging support, encrypted transport).

The architecture remains technology-agnostic at the design level, while the stack listed above represents the current implementation choices used to deliver the complete system scope.

Security and Risk Considerations

SecureWebOps is designed with a strong emphasis on security due to its handling of sensitive scan data, user authentication information, and security reports. The system applies layered defensive controls, zero-trust principles, and secure data handling practices to protect confidentiality, integrity, and availability across all workflows.

Threat Model Diagram



The SecureWebOps threat model is derived directly from the system's Data Flow Diagrams (DFD Level 1 and Level 2). Each external entity, process, data store, and trust boundary identified in the DFD is evaluated for potential security threats. The threat model focuses on how data flows through authentication, scan execution, reporting, and storage components, and identifies risks using the STRIDE methodology.

DFD Elements Considered

- External Entities:
  -E1: End User
  -E2: Target Website / Email Source

- Processes:
  -P1: User Authentication & Authorization
  -P2: Vulnerability Scan Execution
  -P3: Reporting & Audit Logging

- Data Stores:
  -D1: User & Role Data

-D2: Scan Results & Audit Logs

Trust Boundaries

The following trust boundaries are defined based on the Data Flow Diagram:

- Between E1: End User and P1: User Authentication & Authorization
- Between E1: End User and P2: Vulnerability Scan Execution
- Between System Processes (P1, P2, P3) and Data Stores (D1, D2)
- Between P2: Vulnerability Scan Execution and E2: Target Website / Email Source

These boundaries represent transitions between trusted and untrusted components and are key areas where security controls are enforced.

STRIDE Category	**Threat Description**	**Affected DFD Components**	**Impact (CIA)**	**Mitigation**
S – Spoofing	An attacker impersonates a legitimate user to submit scan requests or access reports	E1, P1	Confidentiality, Integrity	Secure authentication, role-based access control, protected session handling
T – Tampering	Modification of scan results or audit logs	D2, P3	Integrity	Write-protected logs, access controls, integrity checks
R – Repudiation	User denies initiating a scan or accessing reports	P3, D2	Integrity	Non-repudiation through timestamped audit logs
I – Information Disclosure	Exposure of sensitive scan findings or user data	P2, D1, D2	Confidentiality	Encryption at rest and in transit, strict access controls
D – Denial of Service	Excessive scan requests overwhelm system resources	P2	Availability	Rate limiting, request throttling, controlled scan execution
E – Elevation of Privilege	Low-privileged user gains	P1, D1	Confidentiality, Integrity	Least-privilege role

	administrative capabilities			enforcement, authorization checks

Mitigation Techniques

Security Risk	Planned Mitigation Techniques
Unauthorized scan submissions	Secure authentication, request validation, rate limiting
User impersonation	Role-based access control, secure session management
Scan data leakage	Encryption, restricted access policies
Log tampering	Append-only audit logs, integrity protection
Denial-of-service attacks	Throttling, controlled scan scheduling
Privilege escalation	Least-privilege roles, authorization enforcement

Security Principles Incorporated

Principle	How It Is Enforced in SecureWebOps
Least Privilege	Users are granted only the permissions required for their role
Defense in Depth	Multiple layers of authentication, authorization, encryption, and logging
Secure by Default	Sensitive data is protected automatically without user intervention
Zero Trust	All requests are validated regardless of source
Non-Repudiation	Audit logs capture user actions with timestamps
Data Minimization	Only essential data is stored, and sensitive data is protected

Security Controls Summary

Control Type	Description
Authentication	Secure credential verification and session handling
Authorization	Role-based access control across all system functions
Data Protection	Encryption of sensitive data in storage and transit
Database Security	Access-restricted data stores
Logging	Centralized audit logging of security-relevant events
Network Security	Encrypted communication channels

Application Security	Input validation, controlled request handling

SecureWebOps integrates security controls across all DFD-defined processes, data stores, and trust boundaries to ensure the confidentiality, integrity, and availability of system data and functionality.

Alternative Designs

Before finalizing the current SecureWebOps architecture, our team evaluated alternative design options that could potentially support web scanning, encryption, phishing detection, and reporting. Below are three alternatives considered, along with the justification for why they were not chosen.

Alternative Design	Description	Difference from Current Design	Why It Was Not Chosen
Browser-only Scanner	Perform scans entirely inside the browser using JavaScript security libraries.	Would not require a backend server.	Major security flaws: keys could leak, browser cannot ethically crawl entire sites, blocked by CORS, no depth scanning. Not acceptable for OWASP testing.
Cloud-Hosted Scanning via Third-Party API	Use APIs from paid vulnerability scanning vendors.	Removes need for ZAP and backend orchestration.	Expensive and violates data privacy since URLs + results are shared externally. Not aligned with our goal of affordable tools for small orgs.
Single Monolithic Web App (No Extension)	All scans initiated through a website dashboard only.	Eliminates extension and simplifies UI.	Users prefer quick on-page scanning. Extension gives faster workflow and auto-detects URLs. Not chosen to maximize usability and reduce friction.

Progress and Plan for Completion

System Functions and Services Implemented to Date

Release 1 – Core MVP (≈ 30% Completion)

Release 1 represents the minimum viable product (MVP) and establishes the core system architecture, security foundation, and scanning workflow. The following system functions and services have been implemented and validated:

- Browser-Based Scan Initiation

  A browser extension enables users to submit website URLs securely for vulnerability scanning. This functionality aligns with the Client/User Interface layer and supports the Vulnerability Scan Execution process (P2) defined in the Data Flow Diagrams.

- Scan Orchestration and Backend Validation

  Backend logic validates incoming scan requests, enforces authentication checks, and orchestrates vulnerability scans against authorized targets. This directly supports the Processing layer and system design.

- Secure Network Routing

  A reverse proxy enforces encrypted communication and centralized routing between clients and backend services, supporting secure data transmission.

- User Authentication and Role-Based Access Control

  Authentication and role-based authorization mechanisms are prototyped to restrict access to scans and reports, aligning with the User Authentication & Authorization process (P1).

- Secure Data Storage and Logging

  Database structures are implemented to store user accounts, scan results, and audit logs with row-level access controls to enforce accountability and data isolation.

- Architecture, DFDs, and Threat Modeling

System architecture diagrams, Data Flow Diagrams (Levels 0, 1, and 2), and a DFD-based threat model using STRIDE have been completed to guide secure development.

Plan for Completion by Release

Release 2 – Platform Expansion & Multi-Tenant Support (≈ 60% Completion)

Release 2 focuses on expanding the MVP into a fully usable, multi-tenant platform while preserving strong security boundaries.

Planned System Functions and Services:

- Multi-Tenant Organization Support

  Enable multiple companies to use the SecureWebOps platform concurrently. Each organization will have its own isolated tenant with a dedicated set of users, roles, scan results, and access controls.

- Organization-Scoped Role Management

  Implement company-specific roles (e.g., administrator, analyst, user) to ensure users can only access data and functions within their assigned organization.

- Enhanced Dashboard and Reporting

  Expand the dashboard to display scan history, security scores, alerts, and encryption status at the organization level.

- Phishing Detection Module (Initial Implementation)

  Introduce phishing detection for submitted links or files and integrate results into the existing reporting workflow.

- Secure File Upload and Encryption Workflow

  Implement encrypted handling of uploaded reports or artifacts to protect sensitive data.

Planned Responsibilities (Vertical Ownership):

Gavin Moore: Multi-tenant architecture design, secure scan workflows, platform security enforcement

Blossom Anolue: Phishing detection logic and user workflow integration

Drishti Tejwani: Encrypted file handling and backend orchestration

Melinda Ngako: Role enforcement, logging expansion, and compliance controls

Release 3 – Full System Completion & Deployment (≈ 100% Completion)

Release 3 finalizes all remaining functionality and prepares the system for complete deployment and evaluation.

Planned System Functions and Services:

- Complete Multi-Tenant Access Control

  Finalize isolation between organizations to ensure strict separation of users, data, and scan results across tenants.

- Advanced Reporting and Audit Capabilities

  Provide comprehensive reporting with historical analysis, downloadable reports, and detailed audit trails.

- Refined Phishing Detection and Alerting

  Improve phishing detection accuracy and implement alerting mechanisms for high-risk findings.

- System Hardening and Performance Optimization

  Apply final security hardening, input validation, and performance tuning to support concurrent tenants and scan workloads.

- Final Deployment and System Testing

Deploy the full platform to the target environment and conduct end-to-end functional, security, and usability testing.

Planned Responsibilities:

All Team Members: Final integration, testing, validation, and deployment

Gavin Moore: Final security review, threat model validation, and deployment oversight

Milestone Approach

The project follows a structured, milestone-based delivery model. Release 1 (30%) establishes a secure and functional MVP, Release 2 (60%) expands the platform with multi-tenant and advanced user features, and Release 3 (100%) completes system functionality, testing, and deployment. This phased approach enables controlled scope growth while maintaining security, usability, and system integrity throughout development.

Applied Computing / Cybersecurity Principles

SecureWebOps integrates real industry knowledge from multiple prior coursework areas:

Course	**Concept Used**	**How It's Applied**
Web Security	OWASP, XSS, SQLi	ZAP scanner detects OWASP Top 10 vulnerabilities
Cryptography	AES, KMS, key rotation	PDF encryption designed using AES-GCM + envelope encryption
Software Engineering	Use cases, UML diagrams	DFD, context diagrams, decomposition, MVP planning
Networking	TLS/HTTPS, proxy routing	Caddy reverse proxy + secure certificate handling
Database Systems	Schema design, RLS	Supabase used for RBAC + secure scan storage

Cloud Security	Access control, secrets	Zero-trust extension model, backend-only secrets

Cybersecurity Principles Embedded

- Zero Trust Architecture
- Confidentiality via Encryption
- Integrity via Audit Logs
- Availability via rate limiting + scan job queues

GitHub

https://github.com/Blossom-Anolue/SecureWebOpsApp.git