

Max, Gleiberman
Julian, Hernandez
Elizabeth, Hechavarria
Nathan, Mahabeer
4/15/2026

Capstone Project Documentation

1. Executive Summary

The IoT Shield project implements a compact but production-style security monitoring environment for Internet of Things (IoT) devices using a Raspberry Pi lab and a desktop-hosted Splunk Enterprise SIEM. Over six sprints, the team deployed three core components: (1) a Splunk SIEM stack on an Ubuntu desktop with three dashboards and twelve custom correlation searches, (2) an IoT device VM on the Raspberry Pi that simulates a smart thermostat, security camera, and smart door lock, and (3) a Kali-based attack VM that executes mapped campaigns including port scans, brute-force authentication, data exfiltration, web-app injection, and denial-of-service. During the January–April 2026 evaluation window the environment generated tens of thousands of device events and roughly 900 security alerts, validating the Raspberry Pi as a reliable.

2. Problem & Requirements (O1)

Traditional SIEM deployments assume server-class hardware and are rarely optimized for small IoT environments, which limits hands-on experimentation and visibility for low-cost labs. IoT Shield addresses this gap by exploring whether a Raspberry Pi 5 can host virtualized IoT and attacker workloads, forward telemetry in real time, and still support meaningful detection and analysis in a desktop Splunk instance.

Stakeholders & personas

- **Student security engineers / instructors** – need a reproducible IoT cyber range to demonstrate full attack chains, log forwarding, and SIEM use-cases in a constrained environment.
- **SOC analysts (persona)** – represented by the Splunk dashboards and Search & Reporting views that summarize device activity, alerts by type, and campaign-level insights

Functional Requirements

- Simulate three IoT device types (thermostat, camera, door lock) and generate realistic network and auth logs.
- Forward logs from both IoT and attacker VMs through the Raspberry Pi using rsyslog and the Splunk universal forwarder into the `iot_devices`, `security_events`, and `threat_analysis` indexes.
- Implement at least twelve detection rules covering brute force, port scans, data exfiltration, web

injection (SQLi/XSS), privilege escalation, unusual outbound traffic, malware signatures, and DoS.

- Provide three dashboards: Main Operations, Security Alerts, and Threat Analysis.

Non-functional Requirements

- Raspberry Pi must sustain continuous forwarding without saturating CPU/RAM while event volume scales into tens of thousands of events.
- Splunk dashboards must update in near real time and retain historical context for the entire 2026-01-01 to 2026-04-13 window.
- Each executed attack campaign must trigger at least one corresponding alert and appear in the Threat Analysis dashboard.

3. System Overview & Architecture (O2)

IoT Shield follows a three-tier architecture composed of a desktop Splunk SIEM, a Raspberry Pi infrastructure layer, and two virtual machines that simulate IoT devices and adversary behavior. The desktop hosts an Ubuntu VM running Splunk Enterprise 10.0.1, which ingests events into the `iot_devices`, `security_events`, and `threat_analysis` indexes and exposes Main Operations, Security Alerts, and Threat Analysis dashboards.

On the Raspberry Pi 5, one VM (Developer B) runs Ubuntu and three Python-based device simulators representing a smart thermostat, security camera, and smart door lock; the second VM (Developer C) runs Debian with Kali tools including Nmap, Hydra, and custom exfiltration scripts. Logs from both VMs are centralized on the Pi using rsyslog and then forwarded over the Splunk HTTP Event Collector (HEC) and universal forwarder into the desktop SIEM.

The Main Operations dashboard summarizes total event volume per device and over time, confirming continuous telemetry from all three simulators, while the Security Alerts and Threat Analysis dashboards present derived security insights across roughly 900 alerts and twelve labeled attack campaigns.

In addition to the SIEM lab, IoT Shield includes a single-page static website that serves as the public-facing overview of the project. The site is implemented with one HTML entry point (`index.html`), a single 2,026-line CSS file, and six vanilla JavaScript modules that power the hero animation, attack and 'How it works' carousels, FAQ accordion, contact dialog, and sticky section navigation, without any frameworks or build step.

4. Discipline-Specific Depth (O6)

Threat model & risk analysis

The primary assets in IoT Shield are the simulated smart devices, their management services, and the Splunk telemetry pipeline; adversaries are modeled as internet-reachable attacker nodes on the Debian/Kali VM capable of scanning, brute-forcing credentials, exfiltrating data, and exploiting web interfaces. Key attack surfaces include exposed SSH/RDP and forwarding ports on the Raspberry Pi, web-facing APIs for IoT management, and outbound channels used for data transfer. The twelve

detection rules were intentionally mapped to support common kill-chain stages from reconnaissance and initial access through exfiltration and impact.

Controls & assurance

- **Reconnaissance** – firewall and netflow searches detect multi-port scans and unusual outbound connections to non-standard ports.
- **Initial access and auth** – failed login patterns to admin and IoT accounts trigger brute-force alerts and privilege escalation attempts.
- **Application-layer attacks** – web log searches identify SQL injection payloads and XSS scripts in query parameters.
- **Malware and command-and-control** – AV log searches detect blocked malware signatures on devices such as ubuntu-siem, camera-02, and sensor-hub-03.
- **Data exfiltration and impact** – netflow and proxy searches highlight large outbound transfers and sensitive data leaving to file-sharing or cloud-sync domains.
- **Availability** – rate-based web log searches flag abnormally high request volumes consistent with potential DoS behavior.

Security Testing

Each alert type was validated by executing the corresponding attack from the Kali VM and confirming that events appeared first in Search & Reporting and then rolled up into the Security Alerts and Threat Analysis dashboards. This provided end-to-end assurance that log sources, forwarding, correlation rules, and dashboards all functioned as intended

5. Implementation Notes (O2)

Implementation details such as VM images, port-forwarding rules, HEC token configuration, and SSH/RDP access patterns are documented in the IoT Shield configuration tables and connection instructions. These artifacts, combined with the architecture diagrams, form a reproducible recipe to rebuild the lab and extend it with additional devices or attack playbooks.

Front-End Website Implementation

- The website is a static, framework-free front end: index.html, css/style.css, and six JavaScript modules (banner.js, attacks-slideshow.js, faq.js, contact-modal.js, scroll-top.js, section-nav.js) • Layout and visual design use a design-token system and modern CSS (Grid, Flexbox, scroll-snap) for responsiveness and motion, with around 2,026 lines of CSS.
- JavaScript modules are self-contained, use progressive enhancement, and exit early if their target DOM elements are missing; no bundler, npm dependencies, or transpilation are required. • All images for architecture, dashboards, and attack examples live under assets/images/ and are loaded with native loading="lazy" and decoding="async" to keep the single-page site performant.

6. Testing & Evaluation (O3)

Performance / reliability

The Main Operations dashboard shows that over the evaluation window the environment produced on the order of 14k–43k events per run, with all three IoT simulators appearing consistently in the “Top 3 Devices by Event Volume” panel. The steady event timelines and increasing totals demonstrate that the Raspberry Pi forwarder remained stable under sustained traffic and that Splunk reliably ingested and indexed telemetry across multiple sprints.

The static website was verified locally using `python -m http.server` and the provided PowerShell helper script, and then deployed to a static host by uploading the HTML, CSS, JS, and image assets, confirming that no server-side runtime or database is required.

Detection effectiveness

The Security Alerts dashboard reports roughly 900 total alerts between January 1 and April 13, 2026, with port scans, brute-force logins, and large outbound transfers as the highest-volume categories. The Threat Analysis dashboard aggregates these into twelve distinct attack campaigns, confirms that all three IoT devices were impacted, and visualizes alert coverage by kill-chain stage

Evidence from Search & Reporting

Representative Search & Reporting views for each correlation search provide trace-level evidence that individual events and statistics match the expected behavior for brute force, data exfiltration, traversal, malware, DoS, and other attack patterns.

7. Testing & Evaluation (O5)

From a security perspective, the lab is intentionally exposed via controlled port-forwarding rules that restrict SSH, RDP, VNC, Splunk, and rsyslog to specific ranges on the public IP, allowing targeted remote access for experimentation while minimizing unnecessary attack surface. Log forwarding uses rsyslog and Splunk HEC with token-based authentication and JSON source types, ensuring that telemetry can be authenticated and parsed reliably in the SIEM.

Ethically, all attack activity is contained within a dedicated private lab using simulated IoT devices rather than production systems, which reduces the risk of collateral damage and avoids collecting real user data. The project highlights the privacy impact of logs that contain credentials, sensitive content, or network metadata and uses dashboards like “Sensitive Data in Outbound Traffic” to surface such risks explicitly.

The website uses semantic HTML5 landmarks, ARIA attributes for carousels, FAQ accordions, and the contact dialog, and honors prefers-reduced-motion, which improves accessibility for assistive technologies and motion-sensitive users. Because it is a static site with no backend or stored user data, its security footprint is limited to content delivery over HTTPS; the contact form is intentionally client-side-only until wired to a secure backend.

8. Deployment & Operations

Deployment consists of provisioning the Raspberry Pi 5 with Ubuntu and KVM, standing up the

Developer B and Developer C VMs from the documented ISOs, and installing Splunk Enterprise on the Ubuntu desktop VM. Operational access is managed via SSH, RDP, and VNC on predefined port ranges, while Splunk HEC and rsyslog endpoints are configured once and then reused across devices. Day-to-day operation is primarily through the three IoT Shield dashboards and the saved searches listed in the “Searches, Reports, and Alerts” view.

For the website, deployment is as simple as serving the `index.html`, `css/`, `js/`, and `assets/` folders on any static host (e.g., GitHub Pages, Netlify, S3+CloudFront); local preview uses `python -m http.server 8080` or the included `serve-local.ps1` script.

9. Limitations & Future Work

The project confirms that a Raspberry Pi can reliably host IoT and attacker VMs and act as a log-forwarding hub, but it also shows that full SIEM processing, search, and visualization require desktop-class compute and storage. Current attack coverage is limited to twelve detection rules and three simulated IoT devices; future work could expand the device set, add more advanced adversary behaviors, integrate vulnerability scanning, and explore running lighter-weight analytics directly on the Pi. Additional work is also needed to harden remote access, automate deployment scripts, and incorporate continuous security testing tools such as SAST/DAST or SBOM scanning.

10. Appendices

Website Implementation Guide

See *IoT_Shield_Website_Documentation.pdf* for full details on HTML structure, CSS design tokens and breakpoints, JavaScript module behavior, image assets, and steps to extend or maintain the site.