KEYWARDEN

Final Year Project: Project Proposal

GEORGE WILKINSON
MIRCEA NICOLESCU / TAREK GABER
G.WILKINSON2@EDU.SALFORD.AC.UK
@00677611 | CHC119

Table of Contents

Motivation

Secure Shell (SSH) remains fundamental to the remote administration of Linux systems across academia, industry, and personal homelabs. Despite its robustness, the management of SSH keys often becomes a significant operational and security challenge. Problems such as key sprawl, poor key rotation, the continued use of deprecated algorithms, and the lack of auditing capabilities contribute to unauthorised access risks and weak compliance with modern security frameworks (Ross et al., 2020; OWASP, 2024b).

In smaller organisations, educational settings, and individual research environments, SSH keys are frequently stored and distributed manually, or insecure alternative methods are used. This leads to inconsistent security practices, with password authentication often left enabled and little or no record of access activity. The absence of centralised auditing mechanisms makes it difficult to attribute actions to specific users or to detect anomalies. Existing enterprise-grade tools such as Teleport and HashiCorp Vault are very effective but personally I have found too complex, resource-intensive, or expensive for small teams or prosumer / consumer use. (Teleport, 2022; HashiCorp, 2025)

This project is motivated by the personal need for a lightweight, self-hosted solution that promotes good security hygiene and visibility in environments where enterprise tools are impractical. Additional motivation stems from familiarity with solutions such as Vaultwarden, Authentik, and Grafana which function similar and is aimed at the same audience. It aims to demonstrate secure identity and access management (IAM) principles in an accessible way, providing educational value and practical applicability for small-scale deployments.

Problem Statement

Current SSH key management practices in non-enterprise environments are largely manual and ad-hoc, resulting in fragmented security controls. There is no straightforward, open-source platform that enables centralised key storage, automated distribution, and audit logging without significant configuration effort. This gap presents both a security and usability problem for small-scale system administrators and homelab operators seeking a structured yet lightweight solution. (Teleport, 2022)

Aim

The aim of this project is to design, develop, and evaluate a modular, web-based system; Keywarden. Centralising SSH key management and enforcing secure access policies while providing real-time auditing and telemetry. The system will enable administrators to manage user access transparently, automate key distribution to remote servers, and monitor login activity through a single, cohesive platform. By implementing these capabilities in an efficient and self-contained application, the project seeks to demonstrate how secure access control and observability can be achieved without enterprise overheads. (OWASP, 2024a)

Objectives

1. Develop a user and key management API and a basic, minimal frontend by approximately month two, enabling account creation, role-based access, and secure key upload and storage (Minimum Viable Product).
2. Implement access request and approval workflows by approximately month three, allowing temporary or permanent access grants with policy enforcement.
3. Design and deploy a lightweight server agent by approximately month four that automatically updates authorised keys and revokes expired or denied keys within 30 seconds.
4. Integrate telemetry collection from agent, and admin / user dashboards by approximately month five, providing visibility into successful and failed logins, anomalies, and compliance reporting.
5. Conduct testing, evaluation, and documentation by approximately month six to ensure the system is secure, usable, and suitable for inclusion in a professional portfolio.

Optional Objectives

1. Develop a visualisation administrator dashboard from the ground up using FastAPI and JS libraries such as Plotly.js to enable a first party tailored UX.
2. Implement a RAC (Remote Access Control) browser client to allow end-users to access machines though a web frontend without directly delegating the user readable keys.
3. Integrate with existing wallet extensions (Bitwarden) / develop a browser extension to store client keys to streamline key access while utilising the same API as existing frontend.

Methodology

My project will follow an iterative, Agile-inspired development methodology structured around four major phases. Each phase corresponds directly to one or more project objectives:

Research and Design

This phase involves reviewing relevant literature on SSH security, identity management, and existing solutions such as Teleport and HashiCorp Vault. System requirements will be defined and used to produce architectural and database designs. The output will include entity relationship diagrams, API specifications, and an overall system model forming the basis of the implementation phase.

Implementation

The implementation phase is divided into iterative development cycles. The first cycle focuses on creating a FastAPI backend with PostgreSQL and a minimal frontend to manage users, authentication, and SSH key storage. The second cycle adds access request and approval features, implementing role-based and policy-based access control, taking advantage of a 3rd party IdP such as Authentik, and utilising JWT in Keywarden's API. The third cycle delivers the Keywarden Agent, a lightweight (systemd) daemon that synchronises authorised keys with target servers and revokes expired ones automatically. The final cycle introduces telemetry and dashboard features for logging, analysis, and anomaly detection, using technologies such as Prometheus, Grafana, or a custom-built visualisation layer (given excess time). (Authentik. 2025; Bitwarden. 2020; dani-garcia, 2024, RedHat. 2025)

Testing and Evaluation

Testing will be performed continuously using automated workflows in GitHub and / or Gitea CI/CD Actions upon commit. Functional testing will verify all application workflows, while security testing will focus on key validation, revocation, and agent authentication. Usability testing will be carried out with external participants to gather feedback on clarity and design. Performance testing will assess agent responsiveness and system load handling.

Documentation and Presentation

Comprehensive documentation will accompany the final implementation hosted via Webserver, GitHub and Gitea. Technical documentation will include API references, setup guides, and deployment procedures, while user documentation will provide instructions for administrators and standard users. The project will conclude with a presentation and live demonstration, supported by evaluation results and a discussion of limitations and potential future enhancements. (Authentik, 2025; Bitwarden, 2020)

References

Authentik. (2025). *Welcome to authentik | authentik*. Goauthentik.io. https://docs.goauthentik.io/

Bitwarden. (2020). *Bitwarden Public API | Bitwarden*. Bitwarden. https://bitwarden.com/help/public-api/

dani-garcia. (2024, December 10). *GitHub - dani-garcia/vaultwarden: Unofficial Bitwarden compatible server written in Rust, formerly known as bitwarden_rs*. GitHub. https://github.com/dani-garcia/vaultwarden

HashiCorp. (2025). *HTTP API | Vault | HashiCorp Developer*. HTTP API | Vault | HashiCorp Developer. https://developer.hashicorp.com/vault/api-docs

OWASP. (2024a). *Insecure Configurations | OWASP Foundation*. Owasp.org. https://owasp.org/www-project-top-10-infrastructure-security-risks/docs/2024/ISR03_2024-Insecure_Configurations

OWASP. (2024b). *OWASP Secure Coding Practices-Quick Reference Guide*. Owasp.org. https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/

RedHat. (2025). *12.5. Managing Public SSH Keys for Hosts | Linux Domain Identity, Authentication, and Policy Guide | Red Hat Enterprise Linux | 7 | Red Hat Documentation*. Redhat.com. https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/7/html/linux_domain_identity_authentication_and_policy_guide/host-keys

Ross, R., Pillitteri, V., Dempsey, K., Riddle, M., & Guissanie, G. (2020). Protecting controlled unclassified information in nonfederal systems and organizations. *NIST*. https://doi.org/10.6028/nist.sp.800-171r2

Teleport. (2022). *State of Infrastructure Access and Security Report*. https://goteleport.com/static/resources/white-papers/State-of-infrastructure-access-2022.pdf

Gantt Chart

Task	18-Sep	25-Sep	02-Oct	09-Oct	16-Oct	23-Oct	30-Oct	06-Nov	13-Nov	20-Nov	27-Nov	04-Dec	11-Dec	18-Dec	25-Dec	01-Jan
Project Proposal & Planning	🟦	🟦	🟦													
Environment Setup (Docker, DB, CI/CD)			🟦	🟦												
Database Schema & Migrations				🟦	🟦											
User Authentication & JWT/OAuth2					🟦	🟦	🟦									
Core Models (Users, Keys, Servers)							🟦	🟦	🟦	🟦						
SSH Key Management APIs											🟦	🟦				
Access Request Flows													🟦	🟦		
Audit Logging & Security Events																
Frontend Dashboard								🟦	🟦	🟦	🟦					
Testing & QA (Unit + Integration)																
Documentation & Final Report																
Final Presentation & Submission																
Other Deadlines										ISM AS1	AWD AS1 | CI AS1	MEA AS1				

Task	08-Jan	15-Jan	22-Jan	29-Jan	05-Feb	12-Feb	19-Feb	26-Feb	05-Mar	12-Mar	19-Mar	26-Mar	02-Apr	09-Apr	16-Apr	23-Apr	30-Apr
Project Proposal & Planning																	
Environment Setup (Docker, DB, CI/CD)																	
Database Schema & Migrations																	
User Authentication & JWT/OAuth2																	
Core Models (Users, Keys, Servers)																	
SSH Key Management APIs																	
Access Request Flows	🟦	🟦															
Audit Logging & Security Events			🟦	🟦	🟦												
Frontend Dashboard						🟦	🟦	🟦									
Testing & QA (Unit + Integration)									🟦	🟦	🟦						
Documentation & Final Report										🟦	🟦	🟦	🟦	🟦			
Final Presentation & Submission															🟦	🟦	🟦
Other Deadlines												ISM AS2	AWD AS2 | CI AS2			MEA AS2	

Logbook

All logbook entries added on Thursday of each week, due to being a free day.

18/09/25

Week 1 (15/09 – 22/09)
Hours Spent: 15
Tasks Completed:
- Created list of ideas; found most interesting idea
- Began research of technologies and standards relating to auditing and remote access
- Created development Debian virtual machine under Proxmox, reverse proxied to my domain under https://app.dev.ntbx.io
- Initialised Git repository with a FastAPI skeleton
- Started Proposal

Problems Encountered:
- Push mirror from Gitea (self-hosted) to GitHub
- Lack of good sources for problem

Supervisor Feedback:
- Initial idea solid, but scope may be too large.
- Need to seek approval for self-hosting rather than poseidon

Reflection:
- Happy with general idea, seems easy to scale or shrink given time constraints

22/09/25

Week 2 (22/09 – 29/09)
Hours Spent: 12
Tasks Completed:
- Made rough ER Diagram for Database.
- Created Initial Database and CI/CD automation.
- Created simple pytest scripts to pass CI.
- Found good sources for proposal from NIST, OWASP and other similar projects.
- Started Gantt Chart.
- Set up vscode server on virtual machine to develop locally on the VM, rather than remotely.

Problems Encountered:
- Exam timetable not yet updated for Gantt Chart dates.
- Motivation too personal, need to orient around existing problems.

Supervisor Feedback:
- Need to link methodology and objectives closer together.
- Expand proposal aims to be more in-depth
- Add references to motivation
- Talk to other lecturers to seek approval for non-poseidon hosting.

Reflection:
- Codebase is solid so far, with an easy starting point "template" made in Week 1 and 2
- Virtual machine was a good idea, gives me a development environment isolated from everything else, with snapshots if needed.

29/09/25

Week 3 (29/09 – 06/10)
Hours Spent: 8
Tasks Completed:
- Completed Proposal

- Completed Gantt Chart
- Added docker image build to CI/CD pipeline, ready to build a new image every commit after tests.

Problems Encountered:
- May not be able to host using own hardware. Could be an issue.

Supervisor Feedback:
- Proposal draft looks great, just need to add logbook as appendix.

Non-supervisor Feedback:
- Need to write a statement of requirements to self-host, and why I cannot host under poseidon.

Reflection:
- Good automation given first CI/CD pipeline, push mirror seems to be working great.