Project Plan

6 Devils

Threat Shield

Date	**27/02/2025**
Version	**0.3**
State	**Finished**
Author	**Addi Beenen, Nuno Dias, Kamsa Trifonov, Duy Nguyen, Guillaume Collet, Emilia Hansen**

Version history

Version	Date	Changes	State
0.1	21-02-2025	Made a beginning on the project plan;	Finished
0.2	25-02-2025	Expanded on all sections;	Finished
0.3	27-02-2025	Added more sections;	Finished

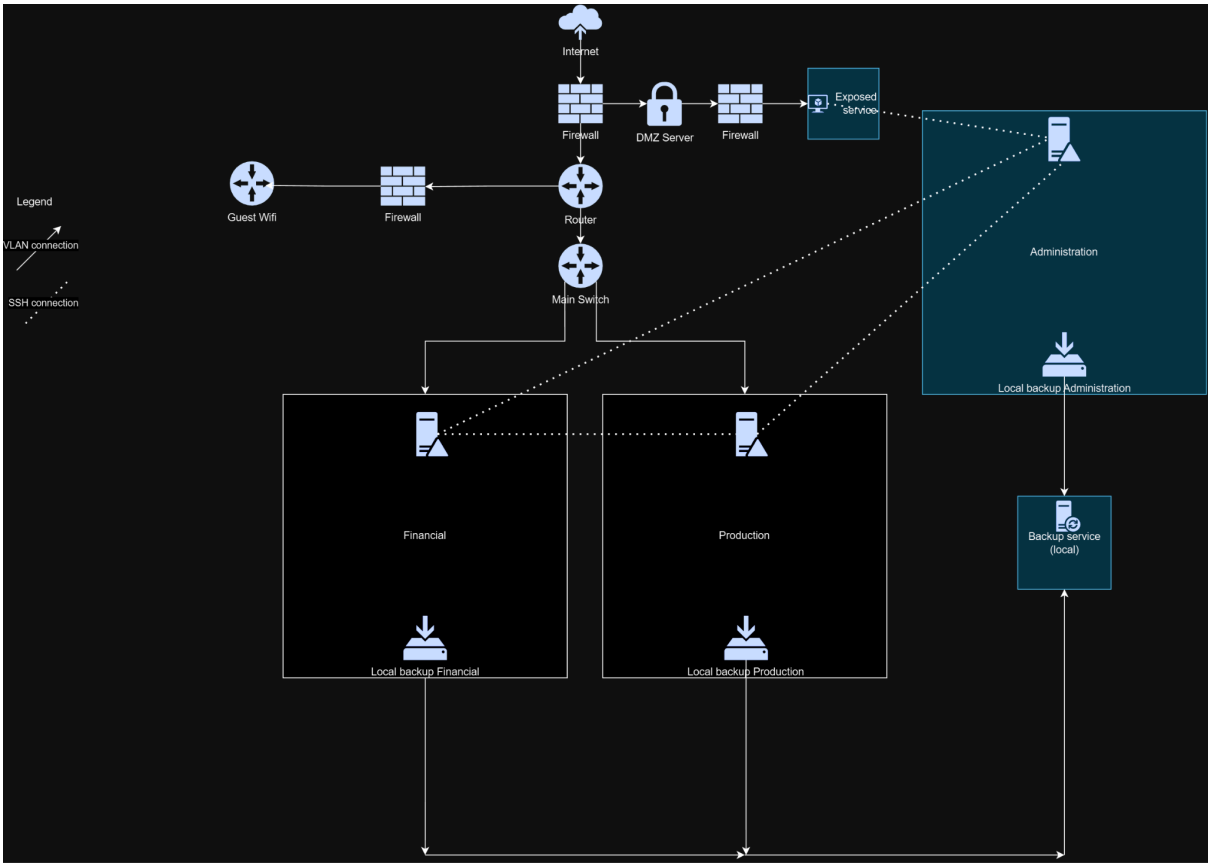
1.Content Table

2.Project assignment

2.1. Context

For this project, our client has an unsecure system requiring securitization. It consists of 3 departments, Administration, Production and Finance. 1 Department is a different physical location. The client has at least 2 separate operating systems running on 2 separate systems, Linux server and Windows Server. What the client expects out of our securitization is segmenting the environments, adding MFA and improving existing authentication and adding a guest/breakout wifi.

Additionally, the client expects the functionality of the system without cloud involvement and additional securitization in the form of point access monitoring, a secure way to host a server to the internet (DMZ). This is the basic environment on which is expected the system to be developed upon. Specific vulnerabilities will be identified as the project progresses. Below is a preliminary network diagram without any network specifics regarding the systems.



The solution they want is a real-time analytics app that will monitor all access to the system and behaviour within the system for anomalous behaviour, access that is not usually used, etc.

This will help to cover blindspots that more traditional solutions don’t notice.

2.2.Problem Statement

A lot of cyber security solutions rely on knowing the exploits that are being used, finding similar patterns and then warning an administrator or isolating the threat. However, this works best when the threats are known, unusual behaviour is not usually flagged. **To be extended.**

2.3.Goal of the Project

The goal of our project is to create a monitoring tool with possibly reactive capabilities. It will function based on log analysis and protect an example network from outside threats preemptively by detecting unusual behaviour.

Other possible avenues include allowing only certain behaviour from certain users. Limiting allowed access, etc. Essentially a zero trust system you need to configure for your whole system.

[...]

From Canvas: “Our solution would essentially be an analysis of logs gotten using a log scrapper.

We will analyze for behaviors and convert the data into human readable information (eg. accessing these folders, logging in at this time, etc.). We will also rank unusual behavior with a score and possibly automate actions for behavior that meets certain conditions, scores, etc.

The Logs will be stored in a third-party solution such as ELK stack or Solr.

Our results will either be displayed in our custom dashboard or feed to another application such as Grafana.”

2.4.Scope

To be determined.

Inside scope:	**Outside scope:**

2.5.Research

Based on the goal and scope previously defined, here is a compiled list of starting questions regarding the necessary research for this project.

1. Storage solutions for large logs files (ELK Stack, Splunk, Apache Solr)
2. Dashboard frameworks for the presentation analytics
3. **[...]**

** More information is available on the results of the research in the research document*

2.6.End Products

[...]
To be determined.

3.Project Organisation

3.1.Stakeholders

Name	Role and Functions	Availability
Marco van der Lee m.vanderlee@fontys.nl	*Role: Tutor*	*In person: Tuesday, Thursday* *Mail: Every Business Day*
Jorg van den Berg jorg.vandenberg@fontys.nl	*Role: Client*	*In person: Tuesday, Thursday* *Mail: Every Business Day*

3.2.Team Members

Name	Speciality	Availability
Addi Beenen 540637@student.fontys.nl	*Software Developer*	*In person: Tuesday, Thursday* *Online: Every Business Day*
Guillaume Collet 572197@student.fontys.nl	*Software Developer*	*In person: Tuesday, Thursday* *Online: Every Business Day*
Nuno Dias n.costadias@student.fontys.nl	*Software Developer*	*In person: Tuesday, Thursday* *Online: Every Business Day*
Emilia Hansen 572565@student.fontys.nl	*Software Developer*	*In person: Tuesday, Thursday* *Online: Every Business Day*
Duy Nguyen 537101@student.fontys.nl	*Software Developer*	*In person: Tuesday, Thursday* *Online: Every Business Day*
Kamen Trifonov 539839@student.fontys.nl	*Infrastructure Engineer*	*In person: Tuesday, Thursday* *Online: Every Business Day*

3.3.Roles

Scrum Master	Guillaume
Communication Master	Kamsa, Addi
Presenters	Kamsa, Nuno, Guillaume
Notetakers	Emilia, Kamsa, Addi

Note: Where there are multiple people the responsibilities will be rotated on a weekly basis

Scrum Master: In charge of making sure the team sticks to our chosen methodology, meetings, deadlines and tools. Leads the retrospective meetings at the end of each sprint.

Communication Master: In charge of external communication with companies, the tutor and the client. Drafts emails, plans meetings and facilitates communication both ways. The bridge between the nerds and others.

Presenter: In charge of reporting, presenting and demoing the team's progress in an engaging manner to any external party. Possibly also in charge of leading said meetings.

Notetakers: In charge of recording key points in meetings as well as structuring and archiving them.

3.4. Documentation (Github)

Github:

- Templates
- Meeting Notes
- Presentations
- Project Plan
- Research Document
- Diagrams

3.5. Methodology

The group will follow an Agile workflow. Tasks will be created, assigned and completed on Trello. Upon completion of a task it must be reviewed by a member not involved in the task before being marked completed.

Research will be documented in the Research Document where choices will be justified and roadblocks will be expanded on.

Meetings:

- Sprint Planning meetings with Tutor and Client (every 3 weeks)
- Weekly stand-up meetings on Tuesdays (internal)
- Client & Tutor meetings based on necessity (every 2 weeks or less)

3.6. Tools

Github Repository - https://github.com/N4fta/threat-shield
For documentation and code. Includes a detailed README with instructions on setup and project structure
Trello - https://trello.com/b/lil7SCjg
For sprint management, keeping track of progress, contribution, etc.
Discord
For internal communication

4.Testing and environment

4.1.Testing strategy

The testing strategy for Threat-shield will focus on ensuring that the system is both functional and secure, this will happen both through manual inspection as well as automated tests.
This will include:

- Testing of sample logs both safe and compromised
- Checking the codebase for common security issues with **Github Integrated Tools**

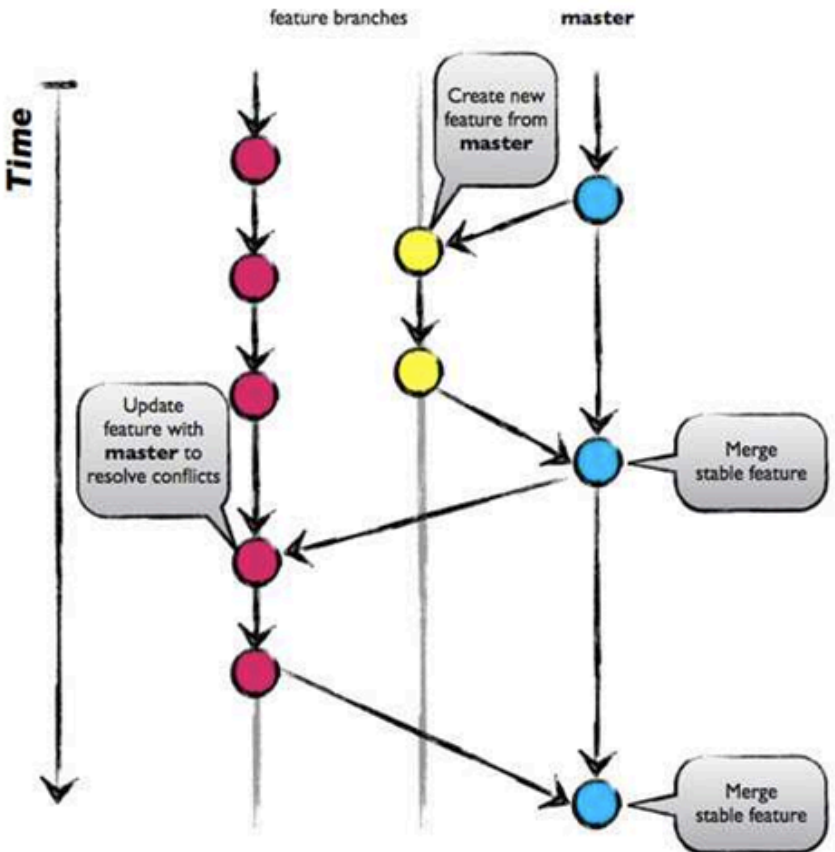
4.2.Test environment

[...]
To be determined.

4.3.Version Management

Threat-shield will use **Git** and **Github** as a versioning system, as well as its integrated tools for CI/CD tasks.

- **Branching Strategy**:
  - A **main branch** will be used for stable releases
  - Tags will be added to stable commits with a version number
  - A **development branch** for ongoing work and feature additions
  - Feature-specific branches for new development, which will be merged into the development branch after passing automated tests



5.Budget

5.1.Human labour

Labourers: 6 junior developers
Hourly Rate: 16 eur/per hour
Estimated human hours per week per person: 20 hours
Estimated cost per week: 1.920 eur
Estimated timeframe in weeks: 16 weeks
Estimated human labour cost: 30.720 eur

5.2.Software Licenses

All the initially planned technologies are open-source projects with open licences for commercial use.

5.3.Hosting

The product will be self-hostable, the only costs will be related to the computer running the program as well as storage solutions for the logs. For storage, depending on the size of the system, amount of logs and amount of backups wanted the space necessary may increase but relative to the upfront cost it is negligible (below 1.000 euros).

5.4.Total estimated cost

Without delays: 30.720 eur + hosting
With 2 weeks worth of extra labour: 34.560 eur + hosting

6.Risk and mitigation

Risk	Prevention activities	Mitigation activities
1. Sickness	Evenly distribute work between members	Online work and redistribution of tasks between other members
2. Data Loss	Version Management with Git and cloud backup off-site with Github	Checking for latest version on team members computers and re-merging
3. Expectations different from result	Frequent updates and sprint-end meetings to align/re-align goals and expectations with client	Reorganizing priorities and planning
4. Scope Creep	Frequent internal meetings (retrospective on sprints) to assess the feasibility of the project, timelines, and goals	Reducing the scope on less critical parts of the project
5. Time Loss	Thorough planning with some breathing room for bug fixing, testing and other problems that can be relied on if needed	Reorganizing priorities and planning