What is SentinelOne and what value does it bring to Q-ICT?

SentinelOne is an EDR and XDR security platform that will protect devices from malware, ransomware, and other kinds of cyberattacks.

EDR (Endpoint Detection Response): is a cybersecurity technology that focuses on detecting and investigating security incidents on endpoints such as desktops, laptops, servers, and mobile devices. It is designed to provide real-time visibility into endpoint activities, detect suspicious or malicious behavior, and facilitate rapid response and remediation.

XDR (Extended Detection and Response): it expands on EDR concept, and integrates new solution like multiple security data sources, including endpoints, networks, cloud environments, and applications, to provide better visibility into security incidents and streamline threat detection and response across the entire IT environment.

SentinelOne has Agents, which is basically an AV (anti-virus) which gives protection against cyber threats. It is a more modernized version of the traditional AV, with the latter having a defined set of vulnerabilities. It utilizes AI and ML to respond to threats in real-time, learning the pattern of an attack in a machine, thus providing behavior-based detection and autonomous protection which the traditional AV (like Avast and Avira) cannot do.

Objectives:

To be able to show the customers their device's status (number of devices protected, endpoints that are infected, get the Site info like last active, device type, resource group it belongs to, device name, OS version, health status, network status, etc.) *See the Sentinel page on SentinelOne dashboard*.

By integrating SentinelOne and having Code Conventions, it will make stronger proof that we are complying with ISO 27000 and NIS2 Directive (an EU guideline to cybersecurity that its member states should adhere to).

What are your desired expectations for the Integration of SentinelOne

3 separate pages

Customer Page:

It should show only the details of their specific devices. A customer cannot see other customers' information about their devices. It should be as user-friendly as possible, implementing graphs, charts, bars, images, and tables to help them visualize. Make it as simple as possible, not overwhelming them, but display enough data to keep them informed and offering them transparency to their security.

This page can be seen on the sidebar of the QaaS app (in between Mijn Dashboard, Mijn Apparaat, etc.,) The purpose of this page is to show the customer what their device's info and status are (based on their Site ID, which is stored in the database in Firestore) and nothing else. No information about other customers should be displayed, or any irrelevant information regarding them. Do not bombard them with too much data.

Pierre: basically, when a customer logs in, he/she only cares whether their device is infected or not. Nothing more.

Helpdesk Page:

Show more technical details than the Customer. Able to see all the Q-ICT customers' devices and their health. The purpose of this page is to provide the Helpdesk with guidance in case of Cyber Incident happens, it should help and guide the Helpdesk o what to do.

This page can be seen in Klanten page, where the helpdesk would need to navigate through all customer and selecting a specific customer, therefore then passing that customer ID and showing to them what SentinelOne can provide regarding their device's information.

Be able to do Analysis Verdict, it will list all the incidents that have happened that have not yet been reviewed, and the Service Desk can mark them as False Positive, Benign, Mitigated, Unsolved, True Positive, Undefined, Suspicious, Not Mitigated, etc. *See Incident page on SentinelOne dashboard*.

Additionally, if a malicious file was detected on a machine, it should send an email notification to alert the selected users and a warning on the screen when the website is open. Optionally, it should also have features like an alert notification sound.

IT (developer) Page:

See the same technical details as the Helpdesk but with more rights to Read and change the Page (Create, Update, Delete?).

Design Choice

Free to choose, do not forget to bear in mind the factor of UI/UX design for the Customers. Look for some ideas online, for example generative AI image generators (like Midjourney)

Stakeholders

Manuel, Robbert, Pierre, and Mark.

What is the next step:

- Implement some design mockups. Do not spend much time on this. Tools that might be used: Adobe XD, Adobe Illustrator, Figma, or Photoshop
- Create UML Diagrams (Class Diagram, ERD for database, Activity Diagram, Relational Diagram, etc., that are necessary for school)
- Together with Manuel, finish and agree upon the Code Convention about Naming Convention (styling), Code Structure, Code Commenting (documentation), Error Handling, Version Control, Dependency Management, Security, Performance Optimization, testing practices, etc. to promote consistency across the codebase (both in the front-end (Flutter) and back-end (Node.js Firebase)). It also helps with code maintenance, scalability, and testability, making it and to collaborate with other developers for better CI/CD pipeline development.
- Together with Manuel choosing the right Design Pattern (e.g., Repository Pattern, Factory, Singleton, Decorator etc.,), and Architectural Pattern (MVC, MVVM, MVP, etc.,) as the general repeatable solution to a commonly occurring problems in programming and software engineering/designing. It will provide a blueprint that the developers can follow and customize to, to solve various problems from simple to complex issues.
- Also look further for more software development OOP (Object Oriented Programming) concepts/ techniques. like ODM and ORM for NoSQL and SQL databases, that provides abstraction to simplify database interactions and helps in mapping objects to Database entities when dealing with large databases.
- Make User Manual (or README.md file) of the application to help the new interns study and understand the code better and faster.
- Implement Unit Testing for Flutter and Node.js (optional, if time permits).
- Can already start on the coding (make some API calls in the back end, show the data in the front-end, choose which graphs or charts are the nicest for the customer to see) and provide MVP (minimal valuable product) for the stakeholder (Robbert) to see.
- Might need to pause/stop on writing the Thesis sub-question about how to integrate SentinelOne to the QaaS app. All the theoretical information has been stated, might be better to continue writing the Thesis after Realization (coding) part has been done.

- Might need another interview after some duration in the Development if any questions/remarks come out or to see how the progress has been made and check if anything is still in order.

Notes

Robert pointed out for me to just take inspiration and reference from the official SentinelOne pages itself. Might need to study and investigate SentinelOne APIs and the documentation further with Manuel ourselves.