Requirements Collection

Genreral requirements

ID	REQUIREMENTS	DESCRIPTIONS
10	Establishing SHES deliverables	Define and document the project deliverables
10.1	Gantt Chart	Diagram representing the project's tasks and their timeline
10.2	Project Charter	Document defining the objectives, scope, stakeholders, and key milestones of the project
10.3	OBS	Organizational structure of the project that determines each person's responsibilities
10.4	RACI Matrix	Table to clarify the roles and responsibilities of each person
10.5	WBS	Breakdown of project tasks into subtasks to facilitate management
10.6	EBIOS RM	Risk analysis method to identify and anticipate potential risks
10.7	Risk Analysis	Assessment of project-related risks, identifying their impact and likelihood of success
20	Using landlock or eBPF	The SuperNanny kernel integration must involve either eBPF and/or LandLock kernel mechanism
30	Coding in Rust	Project development in the Rust programming language
40	Backend development	The SuperNanny backend must be implemented as microservices to ensure scalability, modularity, and maintainability.
50	Microservices communication	Microservices must communicate through standard APIs using a lightweight protocol (e.g., gRPC, REST).
60	Using only Cyber server	The solution must operate entirely within the on-premise data center managed by students in the cybersecurity servers
70	No cloud allowed	External cloud-based services must not be used; all components and dependencies must be self-hosted locally.
80	Microservices language	All microservices must be implemented using the Rust programming language to achieve high performance, memory safety, and reliability
90	Kubernetes usage	Microservices must be deployed on a Kubernetes (K8s) cluster to ensure scalability, fault tolerance, and orchestration.
100	High availability	The deployment must provide high availability (HA) by leveraging Kubernetes features such as: • Multi-replica pods for redundancy. • Automatic load balancing for traffic distribution. • Self-healing mechanisms (e.g., restart failed pods).
200	Stateless microservices	Each microservice must be stateless where possible to support scaling and fault tolerance
300	K8s configs	Kubernetes configurations must include readiness and liveness probes to monitor and restart unhealthy services.
400	Logs and monitoring standard	Logs and monitoring must adhere to the OpenTelemetry standard to ensure unified and vendor-neutral observability
500	Data telemetry	SuperNanny components must emit telemetry data (traces, metrics, logs) using OpenTelemetry instrumentation.
600	Pipeline management	The logs and monitoring pipeline can be managed by the Prometheus suite, including: • Prometheus: For metrics collection and alerting. • Loki: For centralized log aggregation and querying. • Grafana: For visualization of metrics, logs, and dashboards. • Alloy: For telemetry data aggregation and enrichment.

1st semester requirements

ID	REQUIREMENTS	DESCRIPTIONS
10	Detect File Access Requests	As a user, I want SuperNanny to detect when an application tries to access a file (read, write, or execute) so that I am aware of its actions.
20	Notify User on Folder Access Attempts	As a user, I want to receive a real-time notification when an application attempts to access a folder so I can decide whether to allow or block it.
30	Intercept Unauthorized Folder Access	As a user, I want SuperNanny to block a folder access attempt if I choose to deny the request so that my sensitive data remains protected.
40	Detect Network Access Requests	As a user, I want SuperNanny to detect when an application attempts to make a network connection so that I can monitor its behavior
50	Notify User on Network Access Attempts	As a user, I want to receive a real-time notification when an application attempts to make a network connection so I can decide whether to allow or block it.
60	Intercept Unauthorized Network Connections	As a user, I want SuperNanny to block network connection attempts if I deny the request so that unauthorized data transmission is prevented
70	Save User Decisions	As a user, I want SuperNanny to save my decisions (allow/block) for specific applications so that I don’t need to respond to repeated requests.
80	View and Manage Rules	As a user, I want to view, edit, and delete saved rules so I can adjust SuperNanny’s behavior as needed.
90	Allow File Access Based on Patterns	As a user, I want to create rules that allow or deny access to specific file types or directory patterns (e.g., /home/user/secret/*) so I can enforce policies efficiently.
100	Allow Network Access Based on Conditions	As a user, I want to create rules that allow or deny network connections based on IP ranges, ports, or protocols so that I can enforce granular network controls.
200	Monitor and Log File Access Events	As a user, I want SuperNanny to log all file access events for auditing and monitoring purposes.
300	Monitor and Log Network Events	As a user, I want SuperNanny to log all network connection events for future analysis.

2nd semester requirements

ID	REQUIREMENTS	DESCRIPTIONS
10	Configurable Notification Thresholds	As a user, I want to configure thresholds for receiving notifications (e.g., “silent mode” or “only notify for critical resources”) to avoid excessive prompts.
20	Integration with System Tray or GUI	As a user, I want SuperNanny to have a system tray icon or GUI for managing notifications, rules, and logs so that it is easy to interact with.
30	Apply Default Policies for Untrusted Applications	As a user, I want SuperNanny to apply restrictive default rules for newly installed or untrusted applications to enhance security
40	Centralized Policy Management	As an IT administrator, I want to manage SuperNanny policies centrally for multiple workstations so that I can enforce security rules consistently across the enterprise.
50	Monitor Policy Enforcement Status	As an IT administrator, I want to monitor the status of SuperNanny policy enforcement (e.g., compliance, policy violations) across all managed workstations so I can detect misconfigurations or security issues.
60	Push Policy Updates Dynamically	As an IT administrator, I want to push updates to SuperNanny policies dynamically (without requiring manual intervention on workstations) so that security rules can be adapted in real-time.
70	Manage Policies Per Workstation Group	As an IT administrator, I want to collect and aggregate SuperNanny logs (file access, network events) from all managed workstations so I can analyze system-wide activity and identify threats
80	Real-Time Alerts for Suspicious Activity	As an IT administrator, I want to receive real-time alerts when suspicious file or network access occurs on any workstation so I can respond to potential threats immediately.
90	Integration with MDM or Configuration Management Tools	As an IT administrator, I want SuperNanny to integrate with enterprise tools like open source MDM solutions or configuration management tools so I can manage SuperNanny policies at scale
100	Role-Based Access Control for Policy Management	As an IT administrator, I want role-based access control (RBAC) for managing SuperNanny policies so that only authorized users can create, edit, or apply security rules.