Fachhochschule Nordwestschweiz
Hochschule für Technik

# Linux Access Control

## Projektvereinbarung

**IP6 Bachelorarbeit**

Damjan Mlinar

Viktor da Paixão Weilenmann

**FHNW**

Hochschule für Informatik

Studiengang IT

Betreuender Dozent:

Christopher Scherb

Alexander Trapp

Windisch, 18.03.2025

# Inhaltsverzeichnis

# 1 Ausgangslage

In der heutigen digitalen Welt greifen Programme zunehmend auf sensible Systemressourcen wie Dateien, Netzwerkverbindungen oder Peripheriegeräte (Kamera/Mikrofon) zu. Während Linux mit Mechanismen wie AppArmor, SELinux und eBPF mächtige Werkzeuge zur Zugriffskontrolle bietet, zeigen sich drei fundamentale Limitationen aktueller Sicherheitsansätze:

bestehende Regelwerke operieren auf zu grobgranularer Ebene. Sicherheitsprofile gewähren häufig pauschalen Zugriff auf Ressourcen, anstatt einzelne Aktionen zu kontrollieren. Ein Texteditor könnte beispielsweise Schreibrechte für alle Dokumente im Home-Verzeichnis erhalten, obwohl der Nutzer eigentlich nur den Zugriff auf eine spezifische Datei freigeben möchte. Diese mangelnde Präzision steht im Widerspruch zum Grundgedanken des Datenschutzes, der minimale Rechtevergabe erfordert.

Die Konfiguration erfordert tiefgreifendes Expertenwissen. Tools wie eBPF ermöglichen zwar theoretisch feingranulare Filterung von Systemaufrufen, scheitern jedoch in der Praxis an ihrer Komplexität. Die Erstellung sicherer BPF-Programme setzt detailliertes Kernelverständnis voraus – eine Hürde, die für Endbenutzer ohne vereinfachte Abstraktionsebene unüberwindbar ist. Zudem ist die Regeldefinition ausschliesslich vor Programmstart möglich, was dynamische Anpassungen während der Laufzeit verhindert.

Es fehlt es an interaktiven Entscheidungsmöglichkeiten. Kein bestehender Mechanismus informiert Nutzer:innen proaktiv über Ressourcenzugriffe oder ermöglicht kontextabhängige Freigaben. Während moderne Betriebssysteme bei Erstzugriffen auf Kamera oder Mikrofon nachfragen, existiert vergleichbare Funktionalität für Dateisystem- oder Netzwerkoperationen unter Linux nicht. Ein Browser könnte somit unbemerkt auf beliebige Dateien zugreifen, sofern ihm pauschal Dokumentenzugriff gewährt wurde.

Um diese Herausforderungen zu adressieren, sollte ein dynamisches System entwickelt werden, das Berechtigungen ähnlich wie bei Android- oder iOS-Geräten abfragt und verwaltet.

# 2 Aufgabenstellung

Ziel dieses Projekts ist die Entwicklung eines benutzerzentrierten Tools zur feingranularen Ressourcenkontrolle unter Linux. Der Fokus liegt auf:

- Implementierung eines **manuell gesteuerten Supervisor-Tools**, das Prozesse überwacht und Systemaufrufe (SysCalls) mittels seccomp-basiertem RET_Trace-Mechanismus abfängt
- Entwicklung eines intuitiven **User-Tools** zur interaktiven Erlaubnisverwaltung (CLI)
- Analyse und Gruppierung relevanter SysCalls (z.B. Dateizugriff: `open`, `read`, `write`; Netzwerk: `connect`, `socket`)
- Parametrische Regeln (z.B. Erlaube `write()` nur auf `/home/dokumente/*`)
- Kontextabhängige Entscheidungen (z.B. Blockieren von `connect()` an nicht-Port-443)
- Erstellung von 4-5 Demoprogrammen zur Validierung, die gezielt kritische SysCalls auslösen
- Privacy-by-Design-Ansatz: Schutz vor unbeabsichtigtem Datenzugriff statt Abwehr bösartiger Angreifer
-

# 3 Fragestellungen

1. **Technische Machbarkeit:** Lässt sich seccomp mit RET_Trace für Echtzeit-Zugriffskontrolle nutzen,ohne Prozessabstürze zu verursachen?
2. **SysCall-Analyse:** Wie können zusammenhängende Systemaufrufe (z.B. Datei-Öffnen + Lesen) sinnvoll gruppiert werden, um Nutzer nicht mit technischen Details zu überfordern?
3. **Praktische Anwendbarkeit:** Wie lässt sich der Zielkonflikt zwischen granularer Kontrolle und System-Performance/Stabilität lösen?

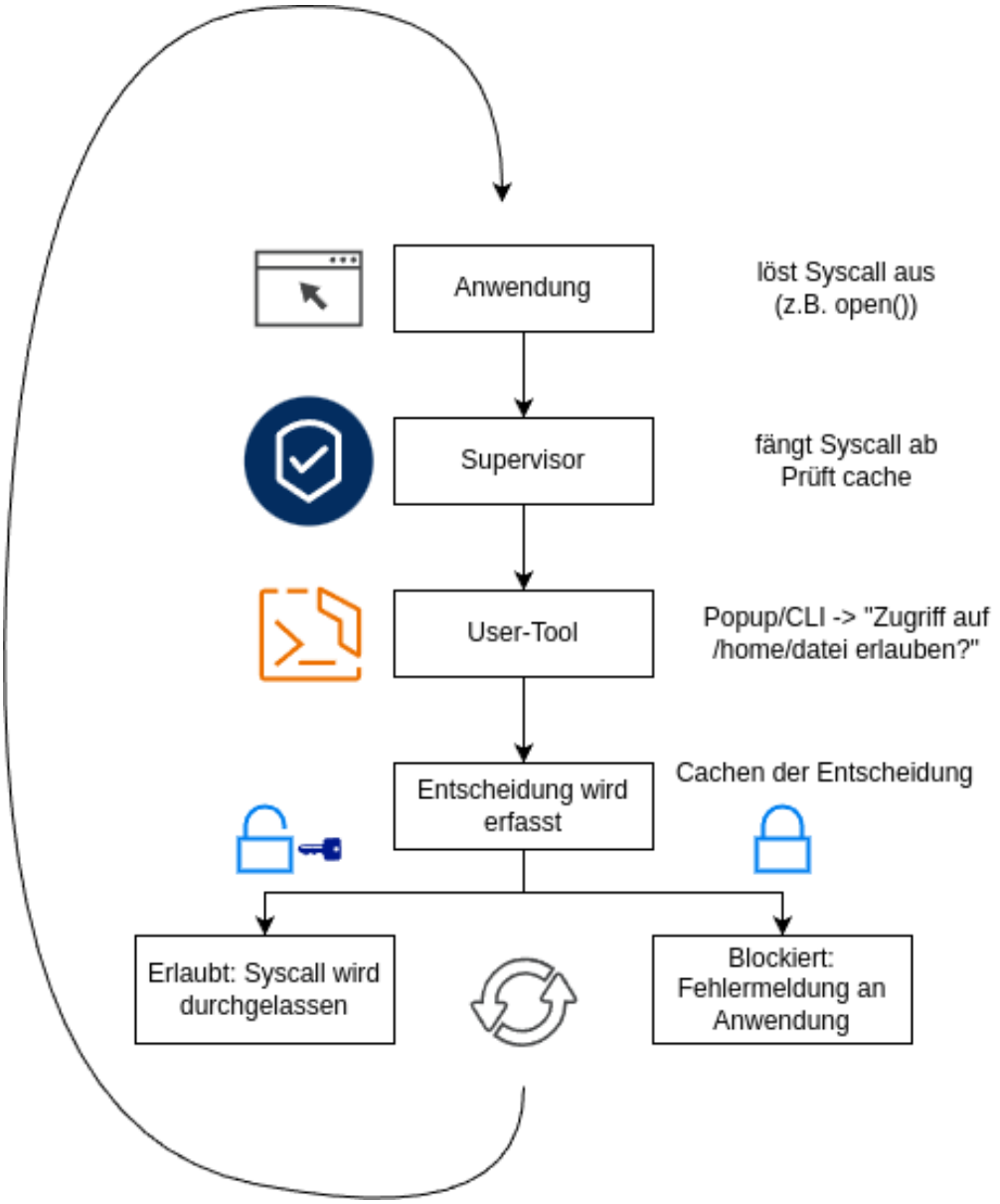
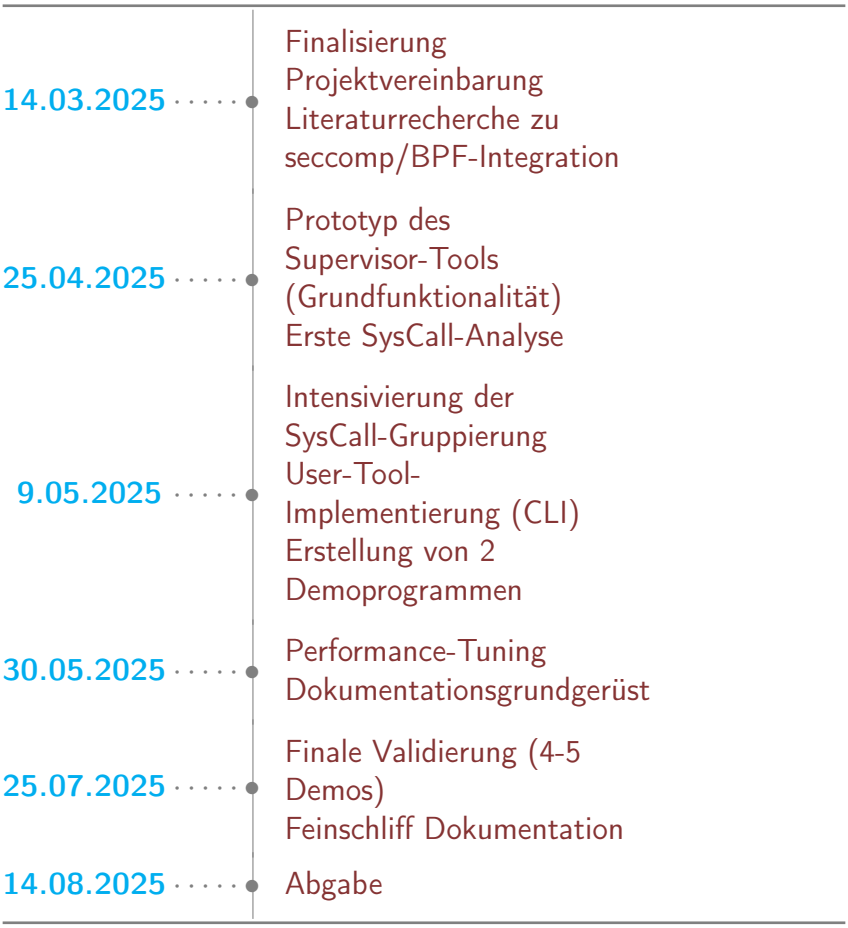



Abbildung 1: User Flow

# 4 Projektablauf und zeitliche Meilensteine




Die strukturierte zeitliche Planung gewährleistet durch monatliche Meilensteine eine kontinuierliche Qualitätssicherung, wobei der letzte Pufferzeitraum gezielt für unvorhergesehene technische Herausforderungen vorgesehen ist.