EDR: Endpoint Agent Modernization & Stalled Project Recovery

Executive summary

Led the modernization of the EDR agent across Windows, macOS, and Linux, resolving long-standing operational issues, improving OS parity, and delivering customer-driven enhancements. Revitalized a stalled agent initiative and delivered a GA-ready release within three months, restoring customer confidence and enabling future roadmap execution.

Context and objectives

- **Product type:** Endpoint Detection & Response (EDR) agent deployed across enterprise environments
- **Business drivers:** Improve stability, OS coverage, supportability, and customer experience; reduce operational friction for SOC and Support
- **Constraints:** Legacy architecture, inconsistent OS parity, certificate lifecycle issues, limited documentation, and a stalled multi-team project
- **Success criteria:** Deliver a GA-ready release, improve OS parity, reduce support escalations, and establish a sustainable modernization path

Role and scope

- **Title:** Product Manager / Product Owner
- **Scope:** Owned product direction, roadmap, backlog, and cross-functional alignment across Engineering, Architecture, SOC, Support, and Customer Success
- **Project length:** Ongoing ownership, with a focused three-month rescue and delivery window for the stalled initiative

Problem statement

The EDR agent suffered from architectural limitations, scalability challenges, inconsistent OS parity, certificate lifecycle issues, and a backlog of customer-requested enhancements. A critical agent initiative had stalled for nearly a year, blocking customer commitments and delaying downstream roadmap items.

Approach and strategy

- **Discovery:** Reviewed architectural designs, support tickets, engineering backlog, SOC feedback, and customer escalations to identify root causes and prioritize fixes
- **Prioritization:** Balanced customer impact, technical feasibility, and risk; focused on addressing scalability, resolving blockers and high-value enhancements
- **Roadmap cadence:** Established a clear sequence for modernization, OS parity improvements, and supportability enhancements

Execution and key activities

- Identified missing requirements and gaps across Engineering, Architecture, and Support that had stalled the agent initiative
- Rebuilt the backlog with clear acceptance criteria, technical dependencies, and sequencing
- Coordinated engineering workstreams across Windows, macOS, and Linux to improve OS parity
- Led certificate lifecycle redesign to eliminate recurring operational issues
- Delivered customer-driven enhancements, including expanded OS support and improved supportability features
- Established a repeatable release readiness process for future agent updates
- Partnered with Support and Customer Success to validate fixes and prepare customer-facing communication

Technical architecture and modernization areas

- **OS coverage:** Windows, macOS, Ubuntu, Debian, RHEL
- **Modernization focus:**
    - Increase scalability
    - Certificate lifecycle redesign
    - OS parity improvements
    - Supportability enhancements
    - Legacy architectural cleanup
- **Operational improvements:** Reduced support escalations, improved agent reliability, and enabled future roadmap execution

Deliverables and artifacts

- Rebuilt product backlog with clear requirements and acceptance criteria
- Updated architecture and dependency documentation
- Release readiness plan and cross-team coordination model
- Customer-facing release notes and upgrade guidance
- Internal training materials for Support and Customer Success

Results and metrics

- **Delivery:** GA-ready release delivered in **three months** after nearly a year of stalled progress
- **Stability:** Improved agent reliability and eliminated recurring certificate-related issues
- **Customer impact:** Delivered long-requested enhancements and expanded OS support
- **Operational impact:** Reduced support escalations and improved SOC visibility into agent behavior
- **Strategic impact:** Unblocked future roadmap items and restored confidence in the agent's evolution

Challenges and mitigations

- **Stalled initiative:** Rebuilt requirements, clarified ownership, and established a milestone-driven plan
- **Legacy architecture:** Partnered with Architecture to identify safe modernization paths
- **OS parity gaps:** Coordinated cross-OS engineering workstreams to align capabilities
- **Support escalations:** Prioritized fixes with highest customer and operational impact

Lessons learned and next steps

- Early cross-team alignment prevents drift and stalls
- Comprehensive requirements discovery to define project completion
- Clear acceptance criteria accelerate engineering execution
- Modernization requires incremental, validated steps rather than large monolithic changes
- Next steps: migration to newer code base, expanded OS support, and enhanced upgrade automation