SPAS · SECURE PLATFORM AUTOMATION SUITE · V1.0

LENNLAY · REFERENCE ARCHITECTURE

Platform Hardening in the Mythos Era.

An AI model now finds in hours what attackers previously needed months to discover. The answer is not new software; it is the platform itself: continuously hardened, auditably automated, permanently standardized.

CLASSIFICATION · MISCONCEPTION CLARIFICATION

- **The Secure Platform Automation Suite (SPAS) is not a tool, not a scanner, not a SaaS.**
- SPAS is a **reference architecture with a principles framework** that transforms platform hardening from a project discipline into a pipeline-driven operational state.
- Value is created through **reproducibility, evidence, and governance**, not through any single product.

EXECUTIVE SUMMARY

- Anthropic Claude Mythos found **181 Firefox exploits** in a single autonomous run on April 9, 2026, a 90-fold increase over its predecessor. Competing models will reach comparable capability within **6 to 18 months**, according to Gartner.
- The prevailing practice of annual penetration testing plus one-time hardening no longer reliably meets the requirements of **NIS2UmsuCG (NIS2 Implementation and Cybersecurity Strengthening Act) and KRITIS-Dachgesetz (Critical Infrastructure Act)**.
- This brief describes the SPAS reference architecture through **7 core capabilities,** a JBoss EAP reference scenario, and a **30-day entry path** with scoping checklist.

As of April 2026 · Version 1.0 · Edition for invited recipients

EXECUTIVE SNAPSHOT

What this brief covers, in four key figures.

181 ×
Firefox exploits in a single run by Anthropic Claude Mythos (April 2026), a 90-fold increase over its predecessor

6–18
Months until competing models reach comparable capability (Gartner forecast)

7
Core capabilities form the operational foundation and the governance layer above it

14 Days
Deadline under KRITIS-Dachgesetz for the initial notification of a significant security incident

CORE ARGUMENT

The prevailing practice - annual penetration testing plus one-time hardening - is no longer sufficient to reliably satisfy audit and supervisory requirements (NIS2UmsuCG, KRITIS-Dachgesetz). What is required is **prevention as a continuous, pipeline-driven process** that generates evidence as a matter of operational routine, not as a quarter-end exercise.

CONTENTS

01 · THREAT MODEL

Mythos has changed the rules.

On April 9, 2026, Anthropic released the autonomous security research tool **Project Glasswing**, built on the **Claude Mythos** model. The documented results fundamentally alter the premise under which platform security has been conceived.

What is documented

- **181 verified Firefox exploits** in a single autonomous run, including 14 previously unknown zero-days. The predecessor found 2 exploits in a comparable run.
- **Autonomous exploit development** from vulnerability discovery to a working proof-of-concept without human intervention.
- **The capability level of highly skilled software engineers** in analyzing attack paths across distributed systems (confirmed by the project paper).
- **According to Gartner's forecast,** competing models from OpenAI, Google DeepMind, Meta, and Chinese labs will reach Mythos-class capability within 6 to 18 months.

Operational implications

The economic model of security pipelines is changing structurally. What previously required significant human effort - vulnerability research, exploit development, chain construction - becomes parallelizable, accelerated, and lower-cost with Mythos-class models. The attacker side gains a structural advantage that defenders operating with conventional methods will find increasingly difficult to offset.

For regulated platforms - banking IT, insurance, KRITIS operators, public sector - this shifts the burden of justification. The prior standard of proof, "CIS benchmark PDF on file, annual penetration test completed," no longer reliably satisfies requirements under NIS2UmsuCG and KRITIS-Dachgesetz under current interpretive guidance.

> **TIME WINDOW**
>
> The realistic preparation window for reorienting platform hardening is 6 to 18 months. Those who begin now can execute the transition in controlled steps. Those who wait risk having to implement it under incident or audit pressure.

01 · THREAT MODEL · REGULATORY CONTEXT

What NIS2UmsuCG and KRITIS-Dachgesetz require operationally.

The German implementation of the NIS2 Directive (NIS2-Umsetzungs- und Cybersicherheitsstärkungsgesetz, NIS2UmsuCG) and the KRITIS-Dachgesetz alter the requirements for affected operators across three dimensions:

DIMENSION	PREVIOUS PRACTICE	REQUIRED FROM 2026
Risk Management	ISMS policy as a document, quarterly review.	Continuous technical and organizational measures at platform level, verifiable from system data.
Reporting Obligation	Informal incident report to BSI.	Structured initial notification within 24 hours, complete notification within 72 hours, final report within 1 month.
Management Responsibility	Security topic delegated to CTO/CISO.	Approval and oversight of risk measures by management, demonstrably, with liability exposure in cases of provably neglected measures.
Supply Chain	Trust-based.	Explicit management of software suppliers, including SBOM evidence and documented lifecycle controls.

The operational break

All four dimensions require **ongoing, technically verifiable evidence**, not one-time documents. From an audit perspective, this directly produces a changed audit culture: auditors and supervisors no longer want to see the PDF on SharePoint; they want the extract from the platform showing that policy was enforced last night - and the night before, and the night before that.

> **CONSEQUENCE FOR PLATFORM STRATEGY**
>
> Platform hardening transitions from a project artifact (produced once, then maintained) to a **continuous, reproducible state** that is generated, verified, and documented from a pipeline. The following chapters describe what that pipeline looks like.

02 · SECURE PLATFORM AUTOMATION SUITE (SPAS)

Seven core capabilities across two layers.

SPAS is not a new scanner generation. It is the operational bracket spanning the existing tooling landscape, making it audit-capable and durably effective for the new requirement environment. The seven core capabilities are organized into two layers: Foundation (01–04) and Governance (05–07).

OPERATIONAL CORE · 01–04

01 · Security Standards, Baselines and Policies

Machine-readable Baselines from CIS, STIG, BSI, and customer-specific requirements, versioned and reusable.

02 · Golden Images and Platform Baselines

Verified, reproducible system images as the defined baseline state for all deployments.

03 · Automation, Deployment and Platform Integration

Pipeline-driven delivery into existing infrastructure, CI/CD workflows, and platform processes.

04 · Compliance Verification and Reporting

Continuous scans with automatically generated audit evidence, no manually maintained compliance reports.

GOVERNANCE · 05–07

05 · Documentation, Traceability and Auditability

Complete, machine-generated evidence from system data, no manually maintained audit document.

06 · Lifecycle and Update Management

Controlled patch, update, and version cycles across all relevant platform components with documented lifecycle controls.

07 · Drift Detection and Enforcement

Automatic detection of deviations from the target state. Drift is reported and corrected rule-based.

The operational core (01–04) continuously delivers policies to the platform: from Baseline to Golden Image, from image to deployment, from deployment to compliance scan. Governance (05–07) makes this process auditable and manageable. Pages 6 through 8 describe each layer in detail.

02 · CORE CAPABILITIES 01–04 IN OPERATION

The operational core: Policy → Golden Image → Deployment → Verification.

Core capabilities 01 through 04 form a continuous flow. Each stage is independently versionable and verifiable; each transition is reproducible and automated. This is the operational foundation without which auditability cannot exist.

01	02	03	04
Policy	**Golden Image**	**Deployment**	**Verification**
CIS, STIG, BSI, customer-specific - versioned as machine-readable Baseline.	Built, signed artifact (VM image, container image, or configuration package) derived from the Baseline.	Rollout via Ansible Collection, CI/CD pipeline, GitOps, or IaC deployer onto servers, VMs, containers, clusters.	Automated compliance scan at configurable intervals. Deviation generates a ticket and report.

What happens at each stage

01 · Security Standards, Baselines and Policies

Customers select the policy profiles relevant to them (CIS, STIG, BSI IT-Grundschutz, custom standards). Golden Images and Baselines are built from these. Policies are machine-readable, versioned, and reproducible. Documentation of deployed profiles follows downstream.

02 · Golden Images and Platform Baselines

A hardened base artifact is built from the policy. For Linux: a VM image or container image. For Windows: a hardened Windows Imaging Format (WIM) or a Desired State Configuration (DSC). For containers and Kubernetes, hardening flows directly into the image build. Every artifact is signed, versioned, and carries a traceable lineage from upstream to production.

03 · Automation, Deployment and Platform Integration

Deployment is the capability most often absent from the debate. Policies and Golden Images alone do not change a production server. For legacy VMs and Middleware: Ansible Collections and CI/CD pipelines deliver the Baseline to the host. For containers and Kubernetes: ArgoCD, Flux, or comparable GitOps tools deploy hardened images into pods and clusters. Manual individual steps represent the critical path at this stage - where they remain, drift remains.

04 · Compliance Verification and Reporting

At configurable intervals, an automated compliance run executes against all target platforms. Reports are generated by default; automatic Enforcement is optionally configurable. Deviation from the target state generates a ticket and is included in the structured report. Audit evidence is thus produced continuously during operations, not two weeks before an audit.

02 · CORE CAPABILITIES 05–07 · GOVERNANCE

Governance: Documentation, Lifecycle and Drift.

Core capabilities 05 through 07 deliver what audit, board reporting, and lifecycle planning require. They are only reliable when the operational path (01–04) runs continuously.

05 · Documentation, Traceability and Auditability

Every change generates an audit entry. Every exception from the target state is recorded with justification and expiry date. The audit report for NIS2UmsuCG and KRITIS-Dachgesetz is generated from platform data, not from an Excel compilation two weeks before an audit.

06 · Lifecycle and Update Management

Support end dates, versions, CVE status, and migration paths are attributes on the asset, not knowledge held in people's heads. For business-critical Middleware such as JBoss EAP, this means concretely: the end-of-support date appears in the central dashboard, the migration pipeline is visible, and the sequence of migrations can be shown to auditors and the board with a single click.

07 · Drift Detection and Enforcement

Between deployments, the actual state of a platform changes: admin interventions, application-layer installations, emergency patches. Drift Detection compares actual state against target state. The result is not necessarily automatic rollback - that would be dangerous for many business-critical platforms - but **transparency**: every deviation is detected, assessed, and either consciously tolerated or controlled remediated.

> **ENFORCEMENT BY RISK PROFILE**
>
> Automatic Enforcement is not appropriate everywhere. For high-availability applications with narrow change windows: Drift is **detected and reported**; remediation proceeds in a controlled manner during the next change slot. Particularly for Middleware such as Tomcat or JBoss EAP, a correction requires a controlled restart within the change window. For stateless infrastructure (container clusters, standardized web frontends), automatic Enforcement can be applied directly.

Comparison: current practice vs. requirement

CAPABILITY	CURRENT TYPICAL STATE	REQUIRED POST-MYTHOS
Drift	Surfaces at the next audit, or not at all.	Detected nightly, ticket automated, Enforcement by risk profile.
Lifecycle	Support end date arrives via memo to Ops; planning exists in a few people's heads.	EOL dates as asset attributes, update pipeline as a visible process.
Documentation	Audit preparation as a major effort two weeks before the review.	Evidence generated during operations; audit report on demand.

02 · MODULE SCOPE

Platform modules in the SPAS portfolio.

Each SPAS module is independently deployable and builds on the seven core capabilities. The typical entry point is the module with the highest leverage, usually a Middleware or Operating System module; additional modules follow incrementally.

MIDDLEWARE — MOST COMMON ENTRY POINT

Secure JBoss EAP Platform Automation

Auditable Baselines, Ansible Collections, lifecycle management for business-critical Middleware.

MIDDLEWARE

Secure Tomcat Platform Automation

Hardened Tomcat Baselines for web frontends and application servers in legacy architectures.

OPERATING SYSTEM

Secure Linux Platform Automation

CIS/STIG-compliant RHEL, SLES, and Debian Baselines with continuous Enforcement.

OPERATING SYSTEM

Secure Windows Platform Automation

Windows Server hardening with DSC/Ansible integration and auditable compliance reporting.

WEB PLATFORMS

Secure IIS Platform Automation

IIS hardening for classic .NET stacks, lifecycle management included.

CONTAINER

Secure Container Platform Automation

Image Baselines, runtime Enforcement, SBOM integration for Docker and Podman environments.

ORCHESTRATION

Secure Kubernetes Platform Automation

Cluster Baselines, policy-as-code, Drift Detection, for on-prem and cloud K8s alike.

PLANNED — PIPELINE

OpenShift · NGINX · Apache · Java Runtime

Additional platform modules in preparation; further platform families available on request.

INTEGRATION WITH EXISTING TOOLS

SPAS does not replace security scanners, SIEM, or ticketing systems. It consolidates their outputs at the platform level and makes them reproducible. Existing investments in tools such as Tenable, Qualys, Splunk, ServiceNow, or Jira can be integrated, not displaced.

03 · MAPPING TO NIS2UMSUCG

Which core capability addresses which NIS2UmsuCG requirement.

NIS2UmsuCG specifies minimum technical and organizational measures (§ 30 BSIG-neu). The seven core capabilities address these requirements at the platform level. The following mapping is intended as a supporting reference for internal gap assessments and does not substitute for legal review.

NIS2UMSUCG REQUIREMENT (§ 30 BSIG-NEU)	PRIMARILY ADDRESSED BY	SUPPORTING
Policies on risk analysis and information system security	CC 01 · Baselines & Policies	CC 05 · Documentation
Handling of security incidents	CC 07 · Drift Detection & Enforcement	CC 04 · Reporting
Business continuity, including backup management and crisis management	CC 02 · Golden Images (reproducible recovery)	CC 06 · Lifecycle
Supply chain security, including security aspects of relationships with direct suppliers or service providers	CC 02 · Golden Images (signed artifacts, SBOM)	CC 06 · Lifecycle
Security in acquisition, development and maintenance of information systems	CC 03 · Automation & Deployment	CC 01 · Policies
Policies and procedures for assessing the effectiveness of risk management measures	CC 04 · Compliance Verification & Reporting	CC 05 · Auditability
Basic cyber hygiene practices and cybersecurity training	CC 01 + CC 03 (versioned Baselines, automated deployment)	CC 05 · Documentation
Policies and procedures regarding the use of cryptography	CC 01 · Baselines (Crypto Policies)	CC 02 · Golden Images (artifact signatures)
Human resources security, access control policies and asset management	CC 06 · Lifecycle & Asset Attributes	CC 01 · Policies
Use of multi-factor authentication	CC 01 · Baselines (Auth Policy)	CC 04 · Compliance Verification

03 · KRITIS-DACHGESETZ & BSI REFERENCE

What KRITIS operators additionally require.

For operators of critical infrastructure, NIS2UmsuCG **and** KRITIS-Dachgesetz apply simultaneously. KRITIS-Dachgesetz focuses on physical protection, resilience, and availability, but likewise requires demonstrable technical and organizational measures.

KRITIS-DACHGESETZ REQUIREMENT	SPAS CONTRIBUTION
Resilience measures, in particular for maintaining operations after incidents	Golden Images enable reproducible platform reconstruction in hours rather than days. Baseline artifacts are signed and available offline.
Incident notification within 14 days as a structured initial report	Compliance verification (CC 04) and Drift Detection (CC 07) deliver structured evidence about the incident perimeter directly from platform data.
Evidence of protective measures to the competent supervisory authority	Documentation (CC 05) and lifecycle reporting (CC 06) generate audit reports without a special project.
Management of third-party dependencies in the supply chain	SBOM integration in the container module; signed Golden Images with traceable lineage.

Reference to BSI IT-Grundschutz

The SPAS Baseline structure integrates the BSI IT-Grundschutz Compendium approach (modules SYS.1.x, APP.3.x, NET.3.x, and others) as versioned, machine-readable policy. The mapping of module to technical control to verification evidence is end-to-end; a Grundschutz check thus becomes a report from platform data.

> **SCOPE BOUNDARY**
>
> SPAS provides no legal advice and no ready-made NIS2 compliance certificate. It provides the technical layer required for internal gap analysis, ISMS audits, and regulatory evidence to become **reliably achievable with reasonable effort**.

Differentiation from SIEM and Vulnerability Scanners

SIEM systems (Splunk, QRadar, Sentinel) observe events. Scanners (Tenable, Qualys, Greenbone) identify known vulnerabilities. Both are necessary but insufficient for the new requirement environment: they produce findings, not the auditable **state** of the platform. SPAS produces the state and thereby provides the reference against which SIEM and scanner findings are evaluated.

04 · JBOSS EAP REFERENCE SCENARIO

How SPAS operates with JBoss EAP: all seven core capabilities.

In many DACH enterprises, JBoss EAP carries business-critical applications: core banking, insurance policy systems, ERP adjacents, public sector portals. Lifecycle pressure (EAP 7 to 8, Red Hat support model) and hardening obligations (CIS, STIG, customer requirements) make this platform the typical entry point.

01 · Security Standards, Baselines and Policies

lennlay draws on a proprietary JBoss hardening policy based on over 20 years of JBoss EAP operational experience and Red Hat best-practice guidance. The policy is versioned and machine-readable for the image build. It is fully configurable: customer-specific hardening requirements are absorbed and integrated.

02 · Golden Images and Platform Baselines

The Golden Image is built on the policy defined in CC 01 and includes JVM configuration and foundational security parameters. For containerized environments, the policy feeds directly into the image build. For legacy deployments on VMs and bare metal, remaining hardening (directory permissions, security realms, TLS) is applied by the lennlay.jboss Ansible Collection at deployment time.

03 · Automation, Deployment and Platform Integration

Configuration is separated from deployment. For VM-based systems (legacy VMs and cloud VMs), the lennlay.jboss Ansible Collection applies application- and stage-specific configuration (datasources, messaging, cluster settings). This allows many applications and stages to be managed using the same Collection with minimal configuration overhead. Ansible is triggered via AAP (Ansible Automation Platform, enterprise), AWX (community variant), or CI/CD tools such as Jenkins or GitLab.

For container environments, a different approach applies: container images carry the policy already in the image build. Rollout is handled by GitOps tools such as ArgoCD or Flux (OpenShift, AKS, EKS, and further Kubernetes distributions) - Ansible does not apply here.

04 · Compliance Verification and Reporting

The lennlay Compliance Scanner checks JBoss EAP instances against the defined policies. It detects running and non-running platform instances and generates structured verification results. Currently, VM-based environments (legacy and cloud) are supported; a solution for container environments is in development.

For reporting, results are centrally processed, parsed, and written to a database. From this foundation, the following are possible: reporting, dashboard visualization, and prospectively AI-assisted analysis.

04 · JBOSS EAP REFERENCE SCENARIO · GOVERNANCE

Core Capabilities 05–07 at JBoss EAP.

05 · Documentation, Traceability and Auditability

Business applications carry historically evolved requirements that not every policy specification can accommodate without review. Typical JBoss EAP exceptions: legacy datasources without TLS transport encryption; security realms with deprecated authentication protocols; EAP subsystem settings (messaging, remoting) that deviate from the Baseline for production reasons; or JVM heap configurations outside the standard range for specific workloads. The policy profile allows explicit, signed exceptions per application or base image, with justification and expiry date. Exceptions appear in the audit report as versioned technical debt and surface automatically at the next lifecycle review. Every configuration change generates an audit entry.

06 · Lifecycle and Update Management

End-of-support and ELS (Extended Life Support) dates per EAP version are visible as asset attributes in the dashboard: which clusters run which version, when regular support expires, which instances are within the ELS window, what the migration status is. The EAP 7 to 8 transition follows a structured migration path: dependency analysis of deployments, Baseline adjustment to EAP 8 standards, test-stage rollout, production migration. For the path from EAP to OpenShift, lennlay containerizes the deployments, transfers EAP configurations into container-native structures, and deploys via GitOps. The sequence of all migrations can be shown to auditors and the board with a single click.

07 · Drift Detection and Enforcement

Between deployments, the EAP state changes: admin interventions, datasource modifications, emergency patches. Drift is detected and reported. For business-critical Middleware: automatic Enforcement requires a controlled restart within the change window, not a blind rollback. Drift reports are forwarded to the SIEM or the ticketing system.

Why this reference scenario holds

All seven core capabilities are exercised completely and end-to-end with JBoss EAP. The resulting artifacts and processes - versioned Baselines, deployment automation, Drift logic, audit schema - are directly reusable for Tomcat, IIS, Linux, Windows, and containers. The first deployment delivers the operational pattern for all subsequent modules.

05 · ENTRY PATH

From Platform Check to Production Module.

STEP 0
Platform Check
15–30 min. online. Assess platform landscape, compliance pressure, and SPAS fit. Free of charge, no pitch.

▶

PRE-STAGE
Onboarding Workshop
2 days, 2 lennlay experts. Assessment of platform landscape, working practices, automation maturity. Flat rate 4,900 EUR.

▶

WEEK 1
Scoping & Baseline Selection
Sharpen platform inventory, fix compliance targets, establish shared target state.

▶

WEEK 2
Module Customization
Baselines tailored to the customer environment, Collections versioned (where Ansible is in use), test stage configured.

▶

WEEK 3–4
Rollout & Audit Report
Production rollout against the defined cluster set. Drift Detection active. Automated compliance report.

WHAT THE SECURE PLATFORM AUTOMATION SUITE CAN DELIVER DEPENDING ON MODULE AND SCOPE

- Versioned Golden Images & Baselines
- Automation components, e. g. Ansible Collections
- Lifecycle overviews for EoS, ELS, and migration status
- Compliance reports with target/actual view and documented exceptions
- Audit evidence and traceable artifact documentation
- Drift reports and integration with ticketing and SIEM processes

Availability and degree of automation depend on the module, the target platform, and the agreed scope. Individual components are still under development for certain platforms.

Scoping Checklist for Internal Assessment

- [ ] Which platform families are in scope (Linux, Windows, JBoss EAP, IIS, Tomcat, Container, Kubernetes, others)?
- [ ] Which Baseline reference is binding (CIS, STIG, BSI IT-Grundschutz, custom standards, customer requirements from business applications)?
- [ ] Which regulatory framework specifically applies (NIS2UmsuCG, KRITIS-Dachgesetz, BaFin circulars, DORA, sector-specific supervision)?
- [ ] Which operating environment: on-premises, cloud (AWS, Azure, GCP), Kubernetes (OpenShift, AKS, EKS), hybrid?
- [ ] What automation already exists (Ansible, Puppet, Chef, SaltStack, in-house scripts)?
- [ ] Which CI/CD and GitOps landscape is operationally integrated (GitLab, Azure DevOps, Jenkins, ArgoCD, Flux)?
- [ ] Which SIEM and scanner investments exist (Splunk, QRadar, Sentinel, Tenable, Qualys, Greenbone)?
- [ ] Which change windows and approval processes must be observed for the initial pilot set?
- [ ] Who forms the internal point-of-contact group (platform lead, security owner, audit liaison)?

Next Step

Schedule a free Platform Check (15–30 min. online) to assess SPAS fit, compliance pressure, and entry potential. Following step: two-day onboarding workshop, flat rate 4,900 EUR.

Schedule Platform Check

QUESTIONS ABOUT THIS BRIEF? CONTACT US!

contact@lennlay.com