SUSE

20 APRIL 2026

Linux Meets Intune

From Enrollment to Enforcement in Himmelblau

Agenda

1. Protocol prerequisites
2. Step-by-step operation sequence
   Discovery -> enroll -> details -> policies -> status -> complianceInfo
3. Packet-capture demo

What is Himmelblau?

Open-source Entra ID + Intune interoperability for Linux

- Himmelblau is an open-source (GPL3+) identity and device management bridge between Linux and Microsoft Entra ID/Intune.

- It enables Linux sign-in and identity resolution through PAM/NSS.

- It supports device registration/enrollment flows and TPM-backed device trust material.

- It integrates with Intune for Linux check-in, policy retrieval, and status/compliance reporting.

- It enforces real policy behavior on Linux: scheduled scripts, built-in compliance checks, and custom compliance scripts.

- Goal: make Linux endpoints first-class citizens in Conditional Access and compliance-driven environments.

Direction: OIDC Integration and Sovereign Cloud Readiness

A practical off-ramp from Entra ID to open, self-hosted identity

- Extend Himmelblau authentication beyond Entra ID to standards-based OIDC providers (for example, Keycloak).

- Preserve enterprise auth expectations during migration: MFA, SSO, etc.

- Reduce cloud lock-in risk by enabling sovereign/private cloud identity backends without re-architecting Linux fleet access.

- End state: Linux endpoints remain managed and policy-aware while identity control shifts to sovereign infrastructure.

Before the first packet

Step 0: prerequisites

Authorization matrix and token audiences

Every operation expects a different audience token:
- Graph discovery: 00000003-0000-0000-c000-000000000000
- Enroll / Details / Policies / Status: 0000000a-0000-0000-c000-000000000000
- complianceInfo (IWService): b8066b99-6e67-41be-abfa-75db1a2c8809

Enrollment

Step 1: service endpoint discovery

Request: GET {Graph}/v1.0/servicePrincipals/appId=0000000a-0000-0000-c000-000000000000/endpoints

Client behavior:
- Graph::intune_service_endpoints(graph_token)
- Parse providerName entries for LinuxEnrollmentService, LinuxDeviceCheckinService, IWService
- Persist URIs in IntuneServiceEndpoints map used by IntuneForLinux

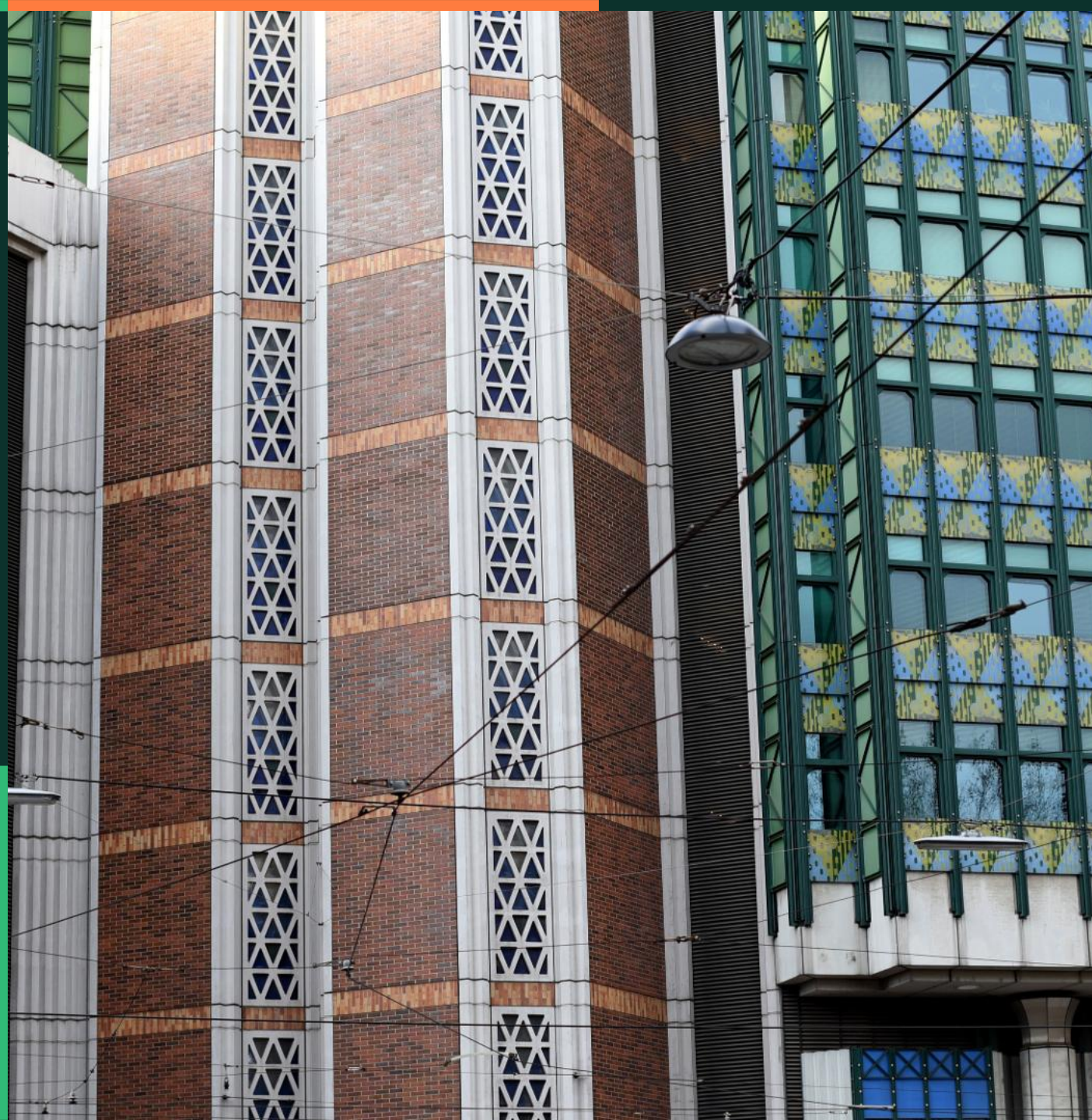
Step 2: enroll

Wire call in libhimmelblau:
POST {LinuxEnrollmentService}/enroll?api-version=1.0&client-version={app_vers}

Request body:
- AppVersion (a recent version string for the MS Intune for Linux app)
- DeviceName
- CertificateSigningRequest (base64 PKCS#10)

Code path:
- IntuneForLinux.enroll(...)
- CSR DER encoded and posted
- certBlob and deviceId returned, tpm associated key

Device Checkin



Step 3: details

Wire call:
POST {LinuxDeviceCheckinService}/details?api-version=1.0&client-version={app_vers}

Payload fields (exact):
- DeviceId
- DeviceName
- Manufacturer
- OSDistribution
- OSVersion

Code path:
- IntuneForLinux::details(...)
- attrs from EnrollAttrs::new(...)

Step 4: policies

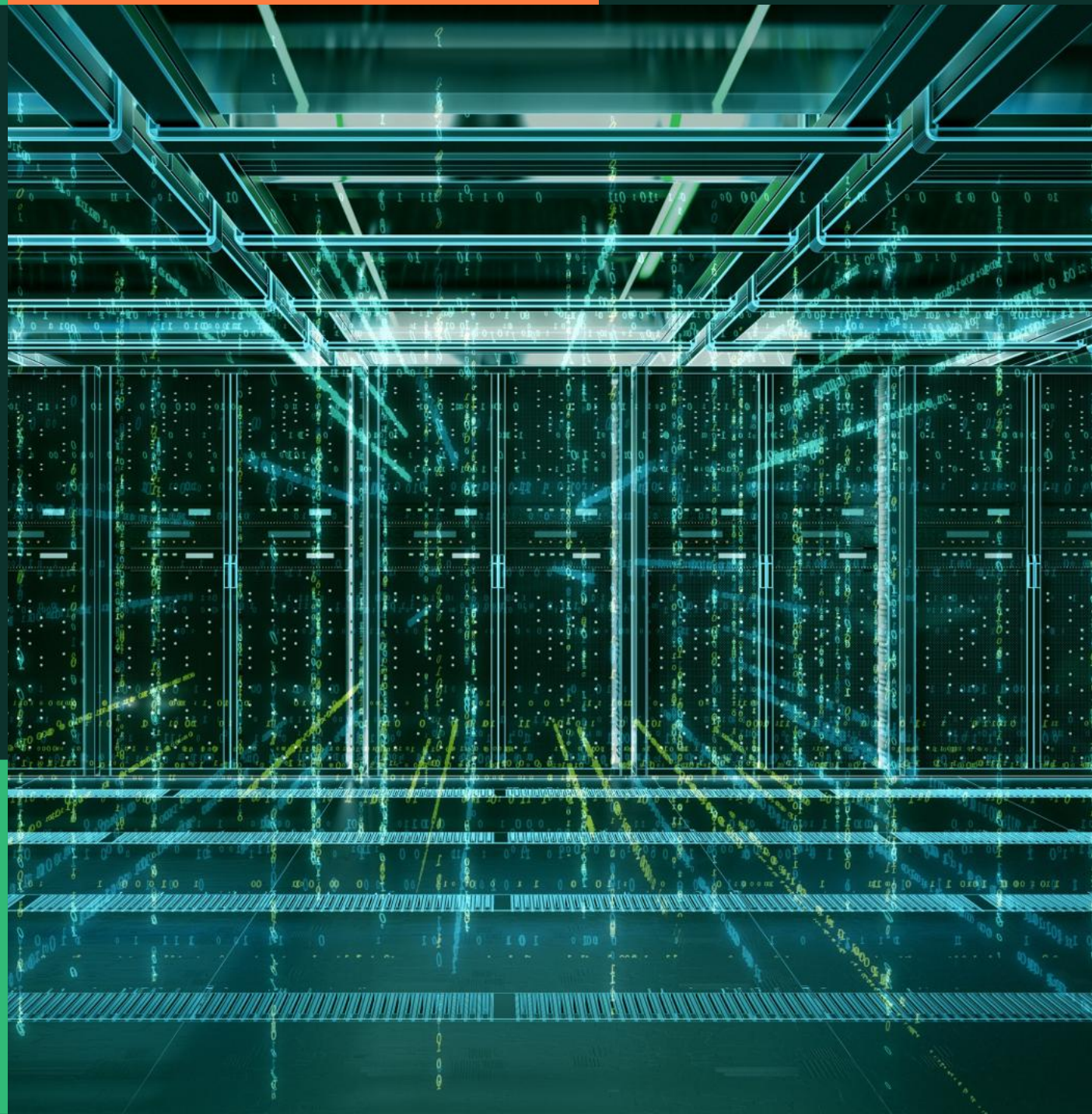
Wire call:
GET {LinuxDeviceCheckinService}/policies/{intune_device_id}?api-version=1.0&client-version={app_vers}

Response model in libhimmelblau:
- IntunePolicyResponse { policies: Vec<IntunePolicy> }
- Each PolicySetting includes ruleId, settingDefinitionItemId, cspPath, value

Himmelblau converts this into IntuneStatus defaults (initially NonCompliant/Unknown per rule).

Himmelblau CSE pipeline

Local policy evaluation



Runtime orchestration in apply_intune_policy(...)

1) resolve endpoints from Graph
2) choose app version via fetch_intune_portal_versions(...)
3) details(...)
4) policies(...)
5) run CSEs over mutable IntuneStatus
6) status(...)
7) sleep(3s), then get_compliance_info(...)

CSE order:
ScriptsCSE -> ComplianceCSE -> CustomComplianceCSE

Scripts Client Side Extension internals

Target rules:
- linux_customconfig_executioncontext
- linux_customconfig_executionfrequency
- linux_customconfig_executionretries
- linux_customconfig_script

Behavior:
- Decode base64 script and write policy_{id}_script.sh + wrapper
- Map frequency minutes to cron schedule (15..10080)
- Write /etc/cron.d/policy_{id}, chmod 644
- Cache per-user applied policy IDs for cleanup

Compliance Client Side Extension internals

Evaluated domains:
- linux_distribution_* (alloweddistros_item_$type | alloweddistros_item_minimumversion | alloweddistros_item_maximumversion)
- linux_deviceencryption_required
- linux_passwordpolicy_* (minimumlength | minimumdigits | minimumlowercase | minimumsymbols | minimumuppercase)

Distribution policies only support Ubuntu and RHEL.
Password policies are irrelevant, given Himmelblau users auth via either Entra ID creds or Hello. We apply them to Hello auth (closest equivalent mapping).

Custom Compliance Client Side Extension internals

Target rule: linux_customcompliance_discoveryscript

Execution model:
- Base64 decode script
- Execute directly if shebang, else via /bin/sh
- Exit code ignored; stdout must be non-empty JSON map
- Success => set_status(..., ComplianceState::Unknown)
- Failure => set_status(..., ComplianceState::Error)

Step 5: status

Wire call:
POST {LinuxDeviceCheckinService}/status?api-version=1.0&client-version={app_vers}

Serialization constraints:
- DeviceId
- PolicyStatuses[]
- PolicyId, LastStatusDateTime, Details[]
- RuleId, SettingDefinitionItemId, ExpectedValue, ActualValue,
 NewComplianceState, OldComplianceState, optional ErrorCode/ErrorType

Step 6: complianceInfo

Wire call:
GET {IWService}/Devices(guid'{device_id}')?api-version=16.4&ssp=LinuxCP&ssp-version={app_vers}&os=Linux&mgmt-agent=mdm

Purpose:
- Fetch service-side ComplianceState and NoncompliantRules

Current Himmelblau behavior:
- Logs compliance state and noncompliant rules

Is this call necessary?
¯_(ツ)_/¯
It's purpose is probably for reporting compliance state to the user (which we don't do).

But why?

- Entra ID determines authentication success/fail based on policy compliance.
- Grace period determined by admin.

Trace-guided walkthrough

Packet capture demo

Demo of Intune packet capture

Known Edge-Cases

- Limitations in the protocol:
  - Only a **single user** can utilize policy enforcement on a device
  - Cannot enroll Linux using Device Enrollment Manager (DEM) accounts

SUSE

Thank You

David Mulder • SUSE

dmulder@samba.org

github.com/himmelblau-idm/himmelblau

intune-spec/intune-spec.md

Questions welcome

SambaXP 2026