Patch it · Fix it · Run it

Patch me, if you can.

Flickst Du noch oder auditierst Du schon?

Christian Stankowic & Jörg Kastning

August 2026

Inhalt

Abschnitt

Wer sind wir und warum sind wir hier?

Jörg Kastning

- Verdiene mein Geld seit 2003 mit EDV/IT
- Seit März 2023 als Mitglied der Red Hat TAM-Community
- Ein TAM ist ein Technical Account Manager
- Benutze Vim; habe Emacs nie verstanden
- Bin ein 1x Engineer, Blogger, TAM-Ambassador, …

Christian ~~Patchowic~~ Stankowic

- IT-Berater und -Trainer bei SVA GmbH
- Linux, Infrastructure as Code, Python
- Patch-Management seit 2014 (Spacewalk)
- sammelt ~~Elektroschrott~~ Retro-Hardware
- Bloggt (cstan.io) und podcastet, u.a.
- Urlaub im Userspace, ThinkPad-Museum
- schreibt fürs Linux-Magazin

Abschnitt

Schwachstellen-Management

Entwicklung der Schwachstellenanzahl

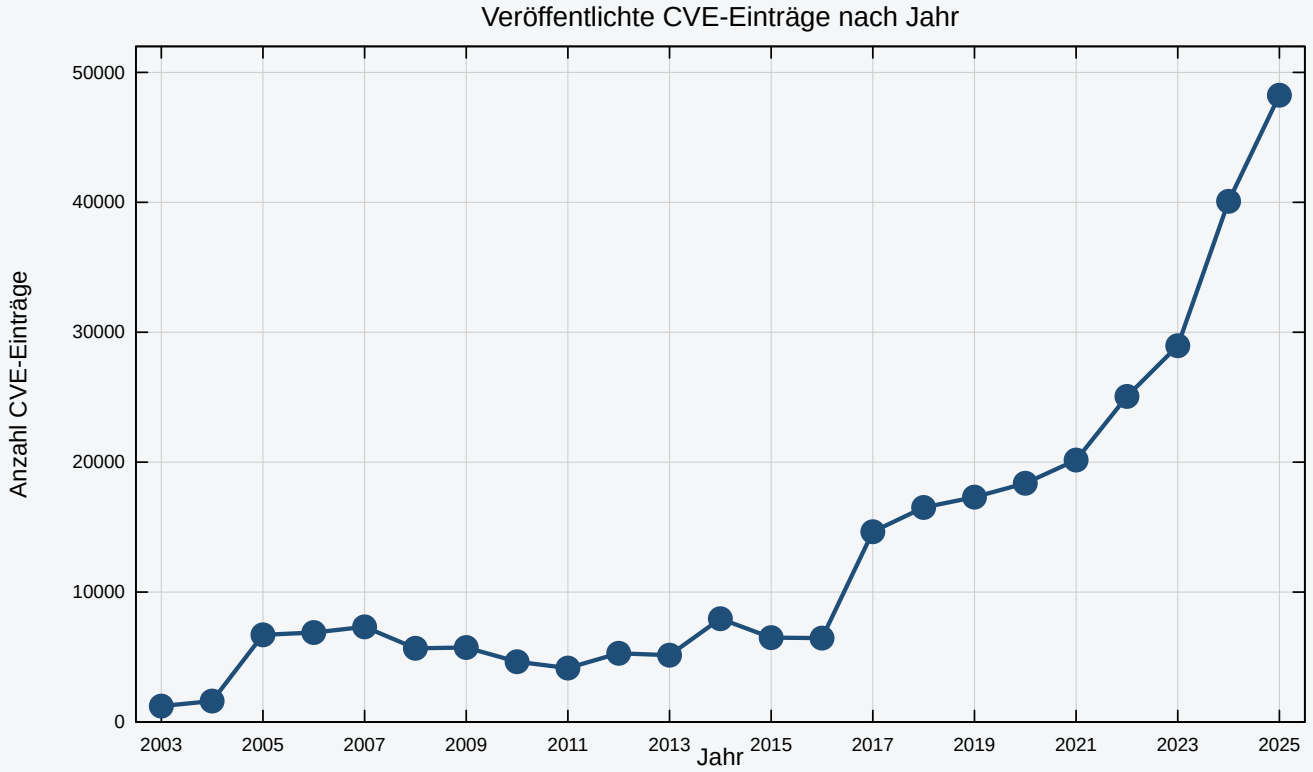
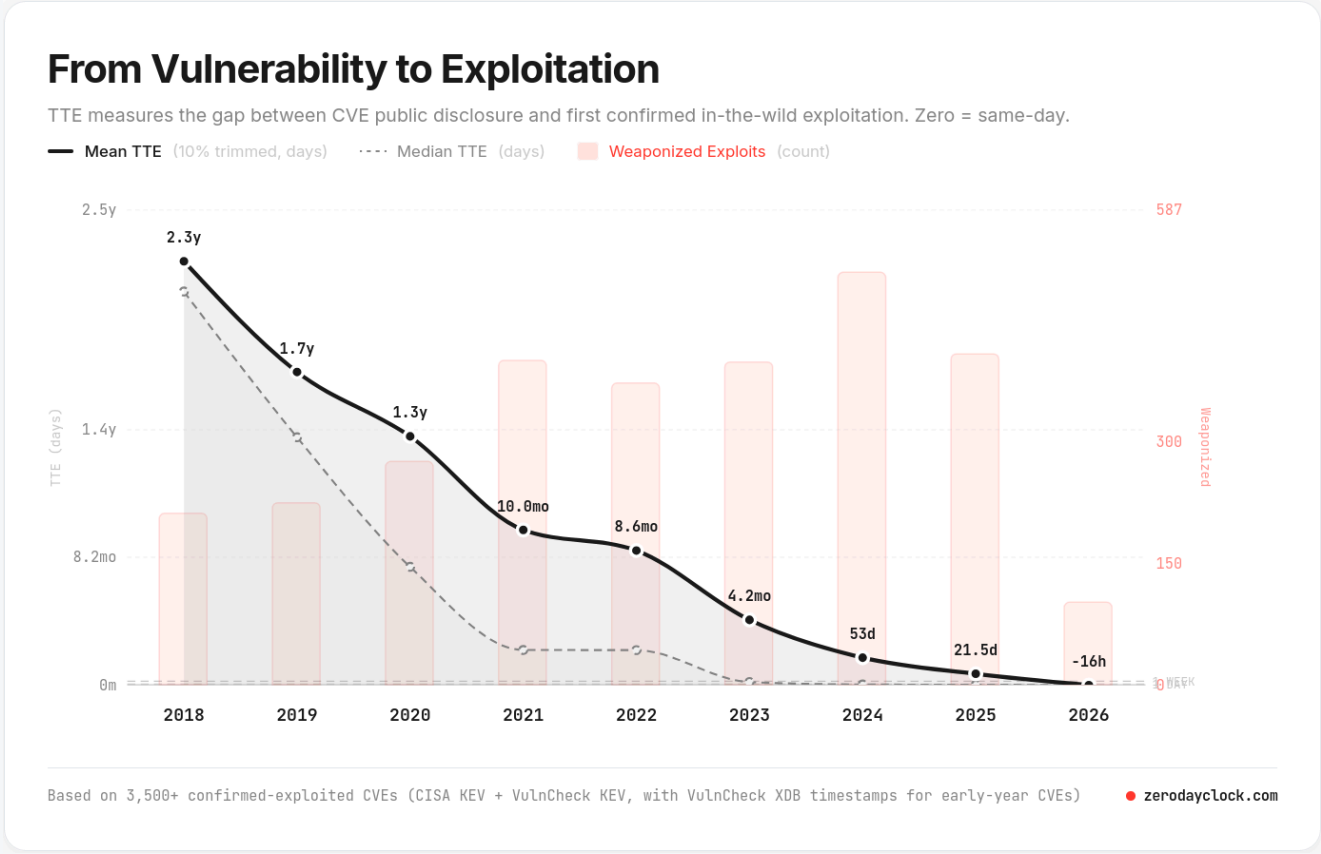


Abb. 1: Entwicklung der Schwachstellenanzahl [1]

Die Uhr tickt



Die Uhr tickt

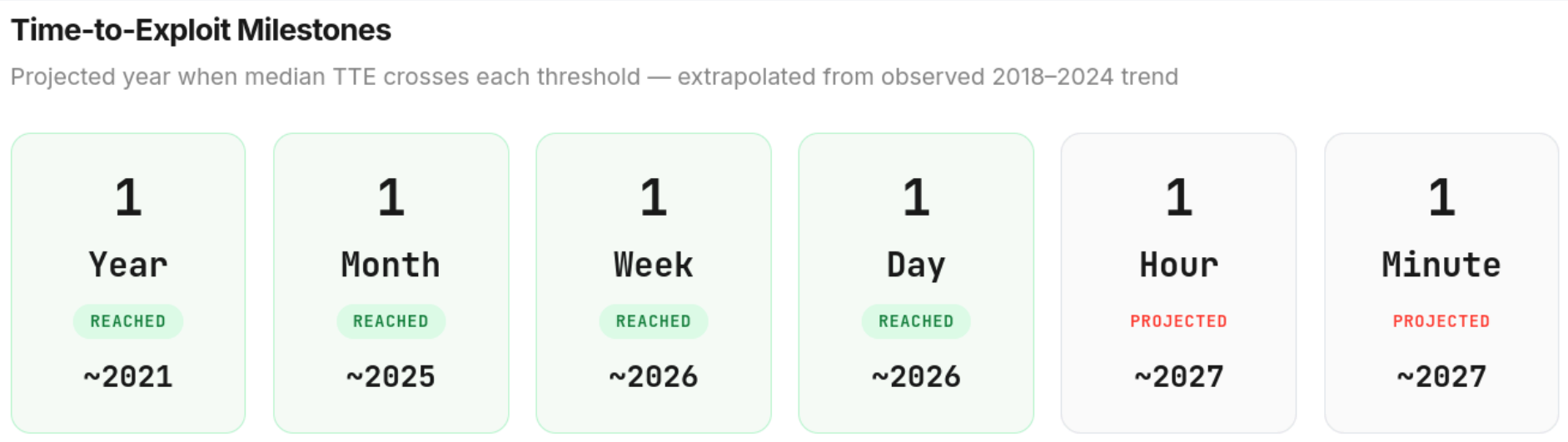


Abb. 2: Diagramme von Zerodayclock [2]

Common Vulnerability Scoring System (CVSS)

Definition und Verweise:

- Der CVSS ist ein allgemeines Bewertungssystem für Schwachstellen [3].
- CVSS gibt es in den Versionen 2, 3 und 4 [4], wobei Version 3 aktuell die gebräuchlichste ist.
- Spezifikation und Verwaltung finden bei der CVSS SIG [5] statt.
- Risiko: Wie warhscheinlich ist es, dass ein Schaden eintritt?
- Schweregrad: Wie schlimm ist es, wenn ein Schaden eintritt?

Common Vulnerability Scoring System (CVSS)

Qualitative Schweregerade nach CVSS v3:

Severity	CVSSv3 Range
None	0,0
Low	0,1-3,9
Medium	4,0-6,9
High	7,0-8,9
Critical	9,0-10,0

Tabelle 1: Abstufungen der CVSSv3-Schweregrade [4]

Common Vulnerability Scoring System (CVSS)

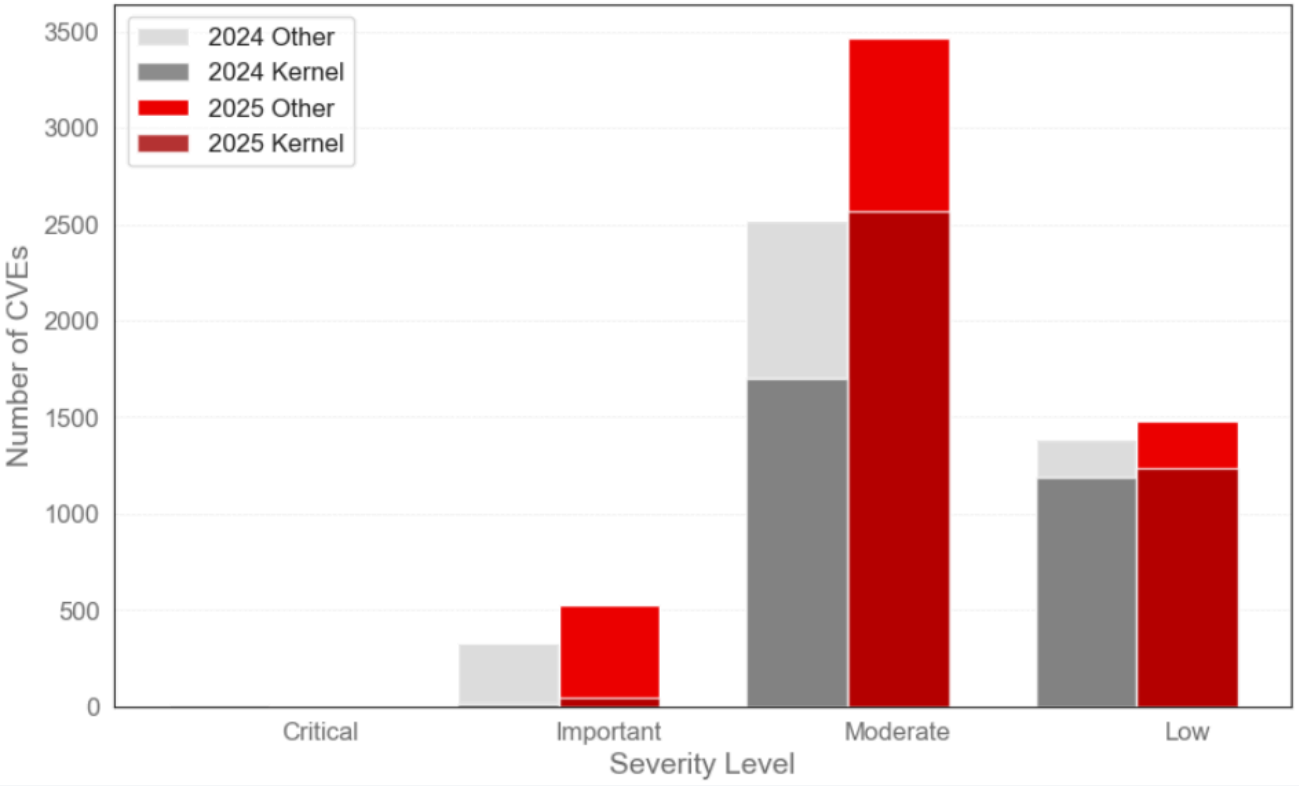


Abb. 3: Entwicklung und Verteilung in den Jahren 2024 und 2025 [6].

Abschnitt

Patch-Management-Konzepte

Herausforderungen und Spannungsfelder

- **Verantwortung:** Wer patcht das Betriebssystem und wer die Anwendung?
- **Erfolgskontrolle:** Funktioniert es hinterher noch wie es soll?
- **Rahmenbedinungen:** SLA, Wartungsfenster, Abhängigkeiten, etc.

Beispiel 1

Phase A	**Warungsfenster planen** 1 pro Quartal; 1 Jahr Vorlauf, meist am Wochenende
Phase B	**Update in DEV/TEST** Installation aller Updates; 2 Wochen Beobachtung, ob es knallt
Phase C	**Update in PROD** Installation der gleichen Updates wie in DEV/TEST

Beispiel 1

Anmerkungen

- Kein Ablaufplan für Notfälle
-
-
-
-
-

Beispiel 1

Anmerkungen

- Kein Ablaufplan für Notfälle
- Kein standardisiertes Verfahren für die DMZ
-
-
-
-

Beispiel 1

Anmerkungen

- Kein Ablaufplan für Notfälle
- Kein standardisiertes Verfahren für die DMZ
- Abhängigkeiten nicht sauber dokumentiert
-
-
-

Beispiel 1

Anmerkungen

- Kein Ablaufplan für Notfälle
- Kein standardisiertes Verfahren für die DMZ
- Abhängigkeiten nicht sauber dokumentiert
- Viele manuelle Schritte
-
-

Beispiel 1

Anmerkungen

- Kein Ablaufplan für Notfälle
- Kein standardisiertes Verfahren für die DMZ
- Abhängigkeiten nicht sauber dokumentiert
- Viele manuelle Schritte
- Hoher Kommunikationsaufwand
-

Beispiel 1

Anmerkungen

- Kein Ablaufplan für Notfälle
- Kein standardisiertes Verfahren für die DMZ
- Abhängigkeiten nicht sauber dokumentiert
- Viele manuelle Schritte
- Hoher Kommunikationsaufwand
- **Ziel:** Automatisierung des Verfahrens; es sollen jeden Tag Updates ausgerollt werden können

Beispiel 1

Erwartungen

Severity	Zbl[1]
None	–
Low	3 Monate
Medium	4 Wochen
High	5 Tage
Critical	48 Std.[2]

Tabelle 2: Vorgabe der IT-Sicherheits-Abteilung zur Installation von Sicherheitsupdates.

[1] Zeit bis zur Installation

[2] Soll auf 3 Std. sinken

Beispiel 2

Phase A

Warungsfenster planen

1 Patchzyklus alle 2 Wochen; alle Nächte und Wochenenden sind genehmigte Wartungsfenster; nur wenige Ausnahmen

Phase B

Installation in TEST

Installation aller Updates, 4 Tage automatisierte Tests, Prometheus-Monitoring

Phase C

Installation in PROD

Installation aller Updates; Monitoring dient zur Prüfung, ob alle Dienste OK sind.

Beispiel 2

Anmerkungen

- Automatisierungsgrad liegt bei ca. 80%, mit Berücksichtigung von Abhängigkeiten, Cluster-Schwenks, Monitoring
- Die 20% Sonderfälle werden manuell oder semi-automatisch (Ansible) aktualisiert
- Beschleunigtes Notfall-Verfahren; Zeit zwischen TEST und PROD auf 1 Tag reduziert
- IT-Sicherheitsabteilung enscheidet, wann ein Notfall vorliegt
- Ist kein Neustart erforderlich, können Updates jederzeit/sofort verteilt werden

Beispiel 2

Erwartungen

- Mehr Test; verbesserte Tests
- Verbesserung Monitoring
- Steigerung des Automatisierungsgrades

Abschnitt

Patch-Management-Werkzeuge

Patch-Management-Werkzeuge

- es gibt eine **Vielzahl** an Open-Source Tools und Suites, die hier helfen
  - von kleinen bis hin zu sehr großen Umgebungen
  - Features variieren, für jede:n etwas dabei 😀

Patch-Management-Werkzeuge

- es gibt eine **Vielzahl** an Open-Source Tools und Suites, die hier helfen
  - von kleinen bis hin zu sehr großen Umgebungen
  - Features variieren, für jede:n etwas dabei 😀
- häufig existieren bereits **proprietäre** Patch-Management-Werkzeuge
  - z.B. für Microsoft Windows-Systeme
  - manche davon haben Linux-Support, i.d.R. ist dieser nicht gut

Patch-Management-Werkzeuge

- es gibt eine **Vielzahl** an Open-Source Tools und Suites, die hier helfen
  - von kleinen bis hin zu sehr großen Umgebungen
  - Features variieren, für jede:n etwas dabei 😀
- häufig existieren bereits **proprietäre** Patch-Management-Werkzeuge
  - z.B. für Microsoft Windows-Systeme
  - manche davon haben Linux-Support, i.d.R. ist dieser nicht gut
- es existieren auch Enterprise-Werkzeuge mit **Support**
  - Red Hat Satellite, SUSE Multi-Linux Manager, ATIX orcharhino
  - zahlreiche zusätzliche Features und wertvolle Integrationen

Patch-Management-Werkzeuge: kleine Umgebungen

	SUSE RMT [7]	Aptly [8]	Repomanager [9]
Erstes Release	2018	2013	2021
Tech-Stack	Ruby, MariaDB, NGINX	Golang	PHP, NGINX, SQLite
Distributionen	SLES, openSUSE Leap	Debian-like	Debian, Ubuntu, RHEL-like
Client benötigt	ja	nein	ja (`linupdate`)
Web-Oberfläche	nein	nein	ja
Staging	nein	ja	ja

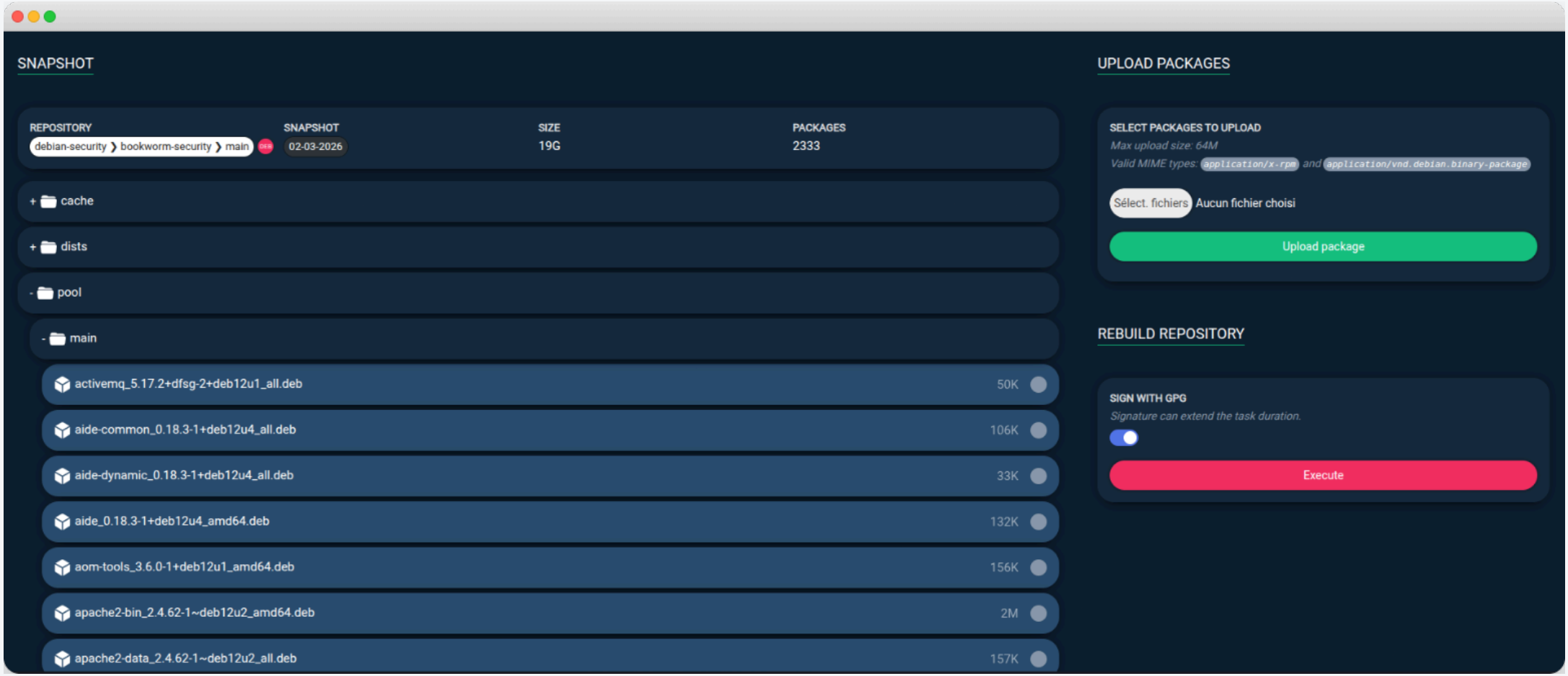


Abb. 4: Über Repomanagers UI lassen sich u.a. Repositories stagen

Patch-Management-Werkzeuge: Suiten

	Foreman [10]/Katello [11]	Uyuni [12]
Erstes Release	2009/2014	2018
Tech-Stack	Tomcat, PostgreSQL, Redis	Tomcat, PostgreSQL, Salt, Podman
Distributionen	RHEL-/Debian-like, Fedora	SUSE, RHEL-like, Debian, Ubuntu
Client benötigt	nein (SSH)	ja (Salt)
Web-Oberfläche	ja	ja
Staging	ja	ja
Configuration Management	Ansible, Salt, OpenVox	Salt, Ansible

Patch-Management-Werkzeuge: Suiten

	Foreman [10]/Katello [11]	Uyuni [12]
Bereitstellung neuer Systeme	PXE, Hypervisor, Cloud	PXE
RBAC	ja	ja
REST API	ja	ja (XMLRPC, JSON)
Automatisierung	CLI (`hammer`), Ansible, Terraform (inoffiziell)	CLI (`spacecmd`), Ansible und Terraform (inoffiziell)
Authentifzierung	Lokale User, LDAP/Microsoft AD-DS	
Security	OpenSCAP-Audits, Mandantenfähig	

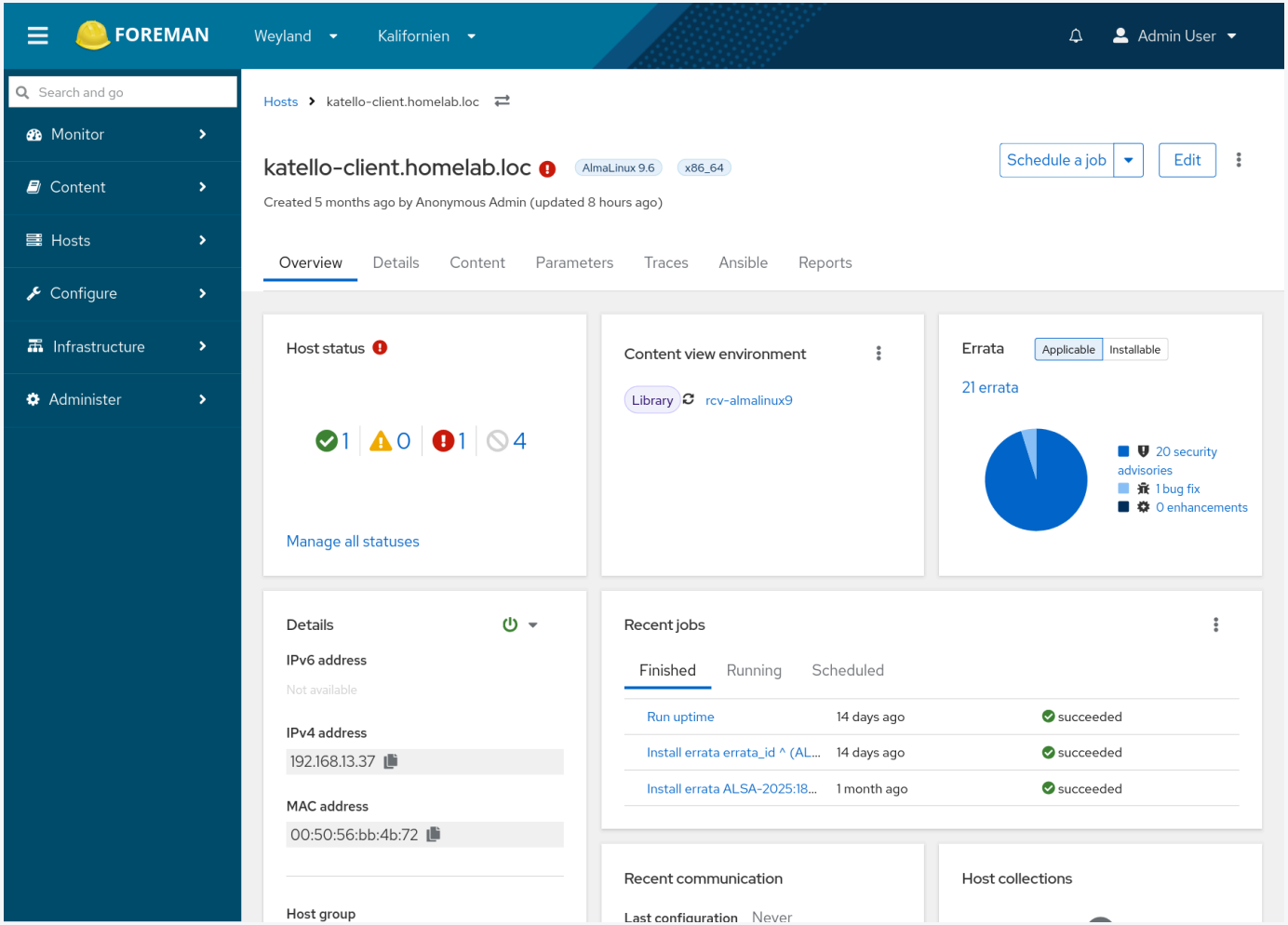


Abb. 5: Ein über Foreman/Katello verwaltetes System

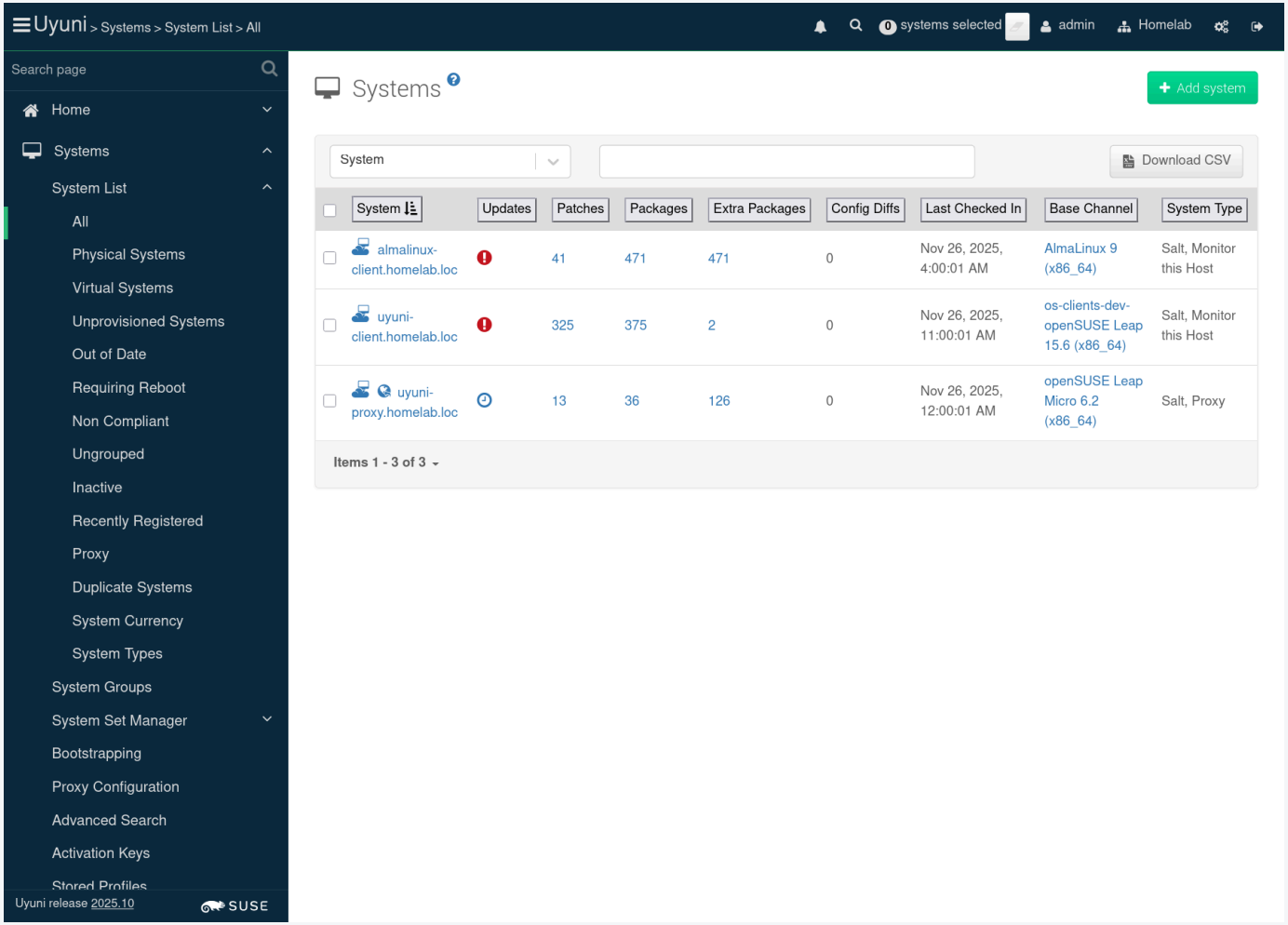


Abb. 6: Über Uyuni inventarisierte Systeme

Patch-Management-Werkzeuge: Enterprise

	Red Hat Satellite [13]	ATIX orcharhino [14]	SUSE Multi-Linux Manager [15]
Basis	früher Spacewalk, Foreman/Katello	Foreman/Katello	Uyuni
Erstes Release	2002	2016	2012
Zyklus	6 Monate	~2 Monate	12 Monate
Wartung	1,5 Jahre	n - 2	3 Jahre
Subskribierung	Managed Systems (RHEL with Smart Management)	Server + Managed Systems (50/1000)	Managed Systems (Lifecycle Management+)
Support-Level	9x5, 24x7		

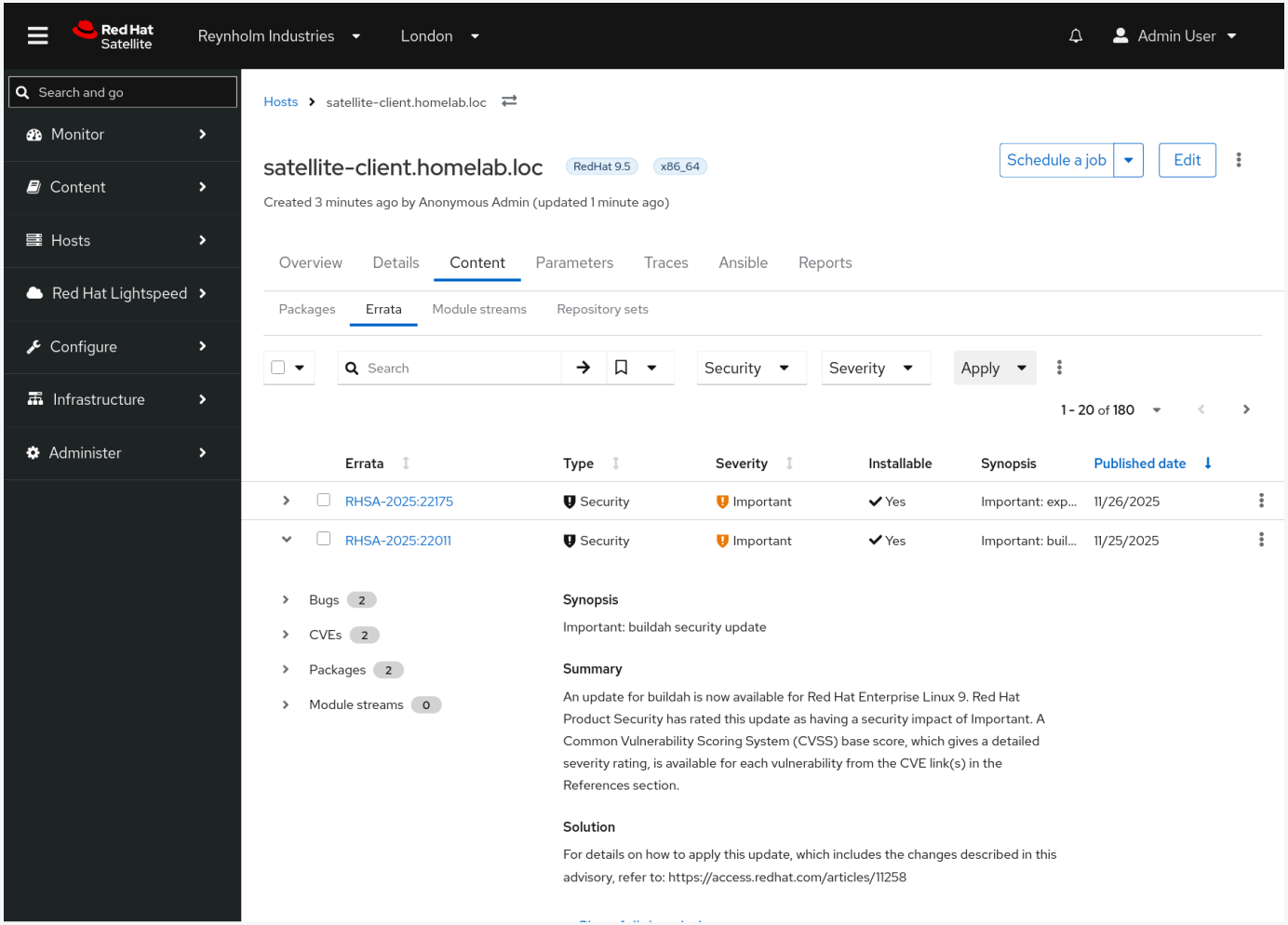


Abb. 7: Red Hat Satellite ähnelt Foreman nicht nur optisch

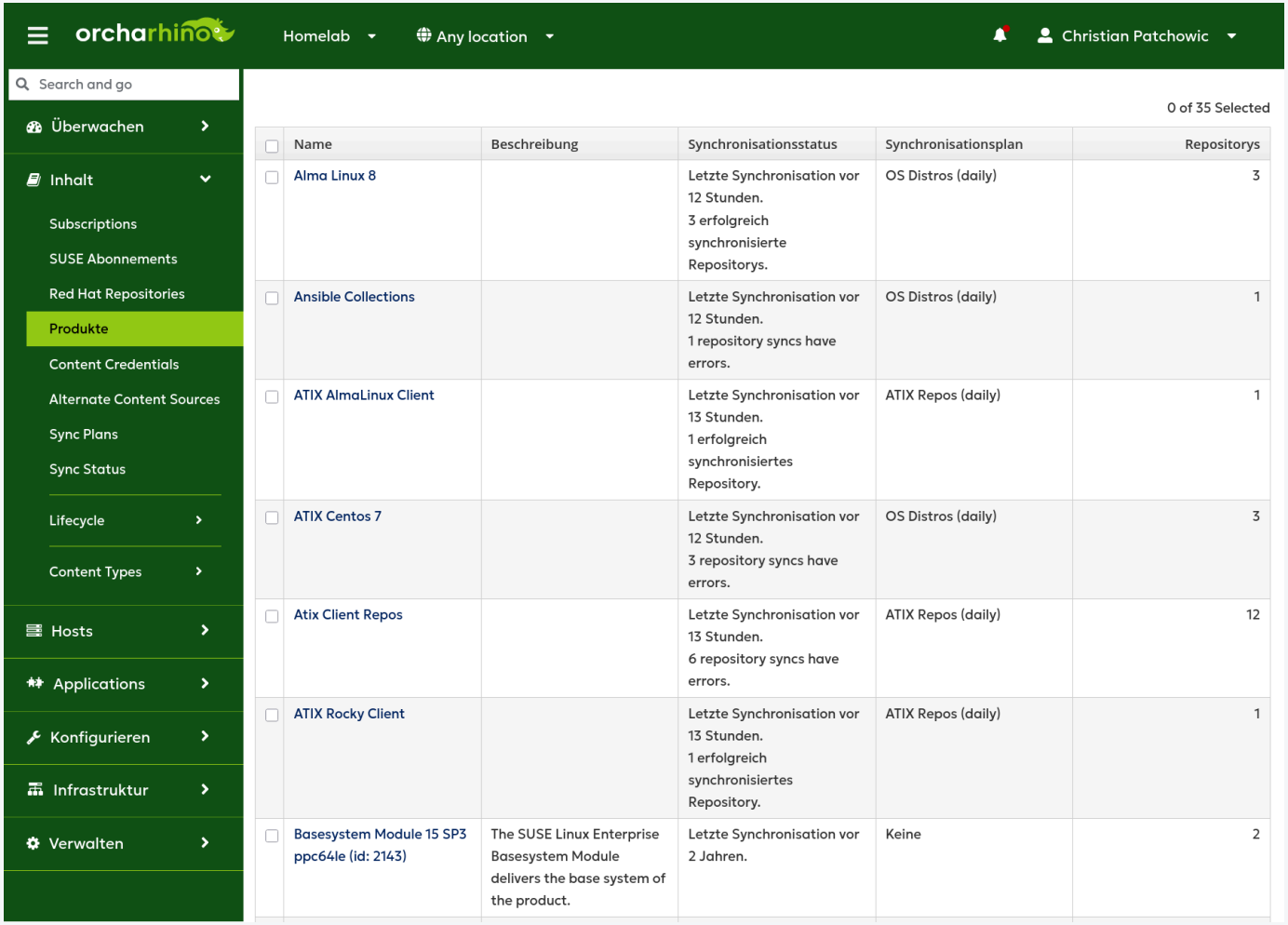


Abb. 8: Orcharhinos technische Unterschiede stecken unter der Haube

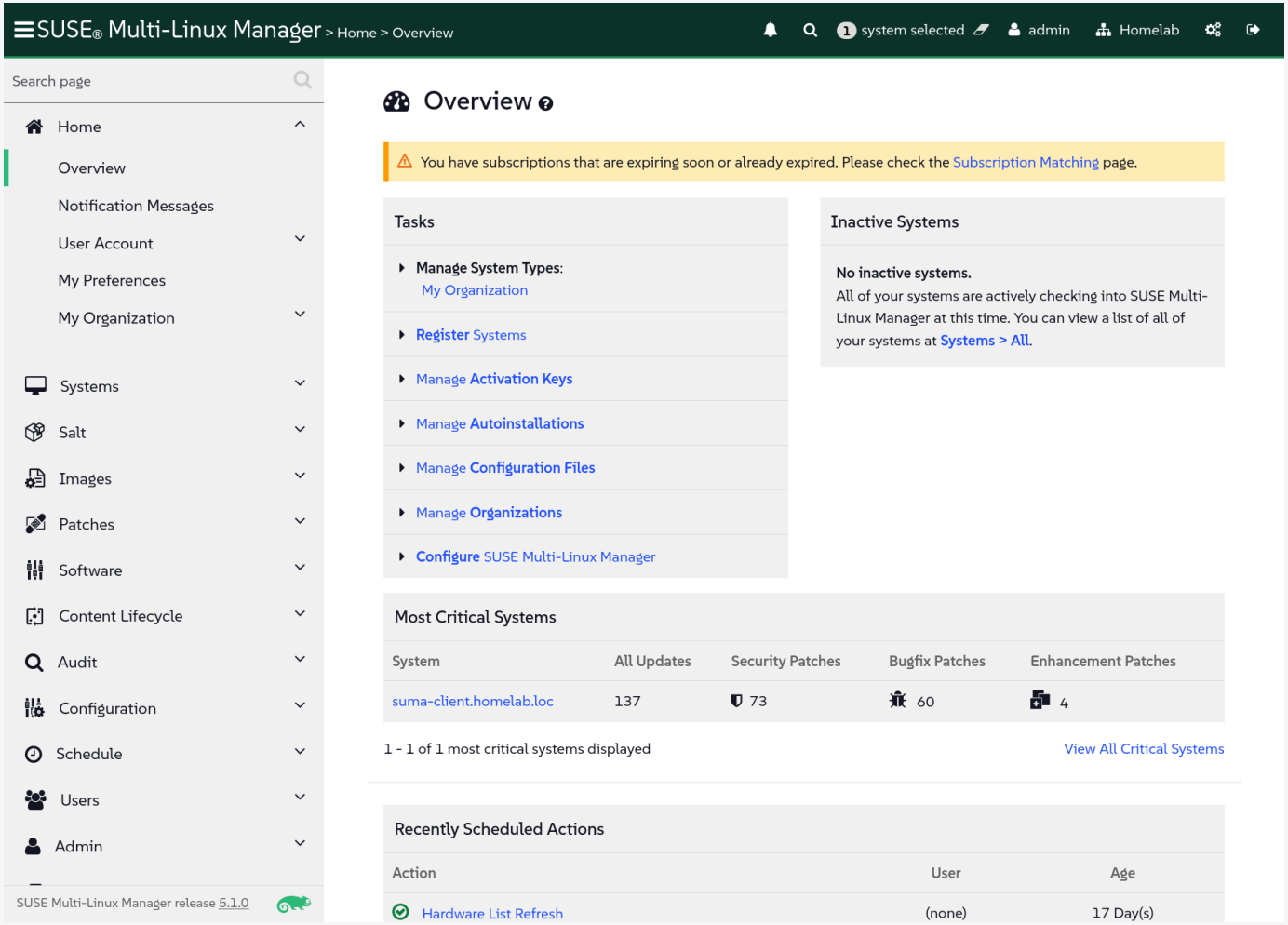


Abb. 9: SUSE Multi-Linux Manager ähnelt Uyuni technisch und optisch

Wann welches Patch-Management-Werkzeug nutzen?

- Homogene SUSE- oder Red Hat-Landschaft
  - SUSE Multi-Linux Manager oder Red Hat Satellite

Wann welches Patch-Management-Werkzeug nutzen?

- Homogene SUSE- oder Red Hat-Landschaft
  - SUSE Multi-Linux Manager oder Red Hat Satellite
- Heterogene Landschaft, viele Distributionen (SLES, RHEL, Ubuntu,…)
  - SUSE Multi-Linux Manager oder ATIX orcharhino

Wann welches Patch-Management-Werkzeug nutzen?

- Homogene SUSE- oder Red Hat-Landschaft
  - SUSE Multi-Linux Manager oder Red Hat Satellite
- Heterogene Landschaft, viele Distributionen (SLES, RHEL, Ubuntu,…)
  - SUSE Multi-Linux Manager oder ATIX orcharhino
- …wird zusätzlich ein hohes Maß an Automatisierung benötigt?
  - ATIX orcharhino

Wann welches Patch-Management-Werkzeug nutzen?

- Homogene SUSE- oder Red Hat-Landschaft
  - SUSE Multi-Linux Manager oder Red Hat Satellite
- Heterogene Landschaft, viele Distributionen (SLES, RHEL, Ubuntu,…)
  - SUSE Multi-Linux Manager oder ATIX orcharhino
- …wird zusätzlich ein hohes Maß an Automatisierung benötigt?
  - ATIX orcharhino
- wenn kein Support gewünscht ist, können auch die Upstream-Projekte verwendet werden
- späterer Wechsel auf Enterprise-Version kann automatisiert werden

Abschnitt

Fazit und Q&A

Fazit und Q&A

- Patch-Management ist **unabdingbar**, insbesondere in Zeiten von LLM-beschleunigten CVEs und Exploits
- Lückenlose **Dokumentation** von Verantwortlichkeiten und Rahmenbedinungen notwendig
- Verlässliches **Monitoring** entscheidend für eine schnelle Prüfung auf Erfolg
- Manuelle Arbeiten (Vorbereitung, Durchführung und Kontrolle) **automatisieren**
- Einsatz eines Patch-Management-Werkzeugs, welches die eigenen **Anforderungen abdeckt**
- Regelmäßige erneute Evaluation des Vorgehens (KISS-Prinzip?)

Vielen Dank.

Fragen?

Quellen und Verweise

[1] “Published CVE Records.” [Online]. Available: https://www.cve.org/About/Metrics

[2] “Zero Day Clock.” [Online]. Available: https://zerodayclock.com/

[3] “CVSS - Wikipedia.” [Online]. Available: https://de.wikipedia.org/wiki/CVSS

[4] “Vulnerability Metrics - National Vulnerability Database.” [Online]. Available: https://nvd.nist.gov/vuln-metrics/cvss

[5] “Common Vulnerability Scoring System SIG.” [Online]. Available: https://www.first.org/cvss/

[6] “Red Hat Product Security Risk Report 2025.” [Online]. Available: https://www.redhat.com/en/resources/product-security-risk-report-2025

[7] “SUSE Repository Mirroring Tool.” [Online]. Available: https://documentation.suse.com/sles/15-SP7/html/SLES-all/rmt-overview.html

[8] “Aptly.” [Online]. Available: https://www.aptly.info/

[9] “Repomanager.” [Online]. Available: https://github.com/lbr38/repomanager

[10] “Foreman.” [Online]. Available: https://theforeman.org/

[11] “Katello-Plugin.” [Online]. Available: https://theforeman.org/plugins/katello/

[12] “Uyuni.” [Online]. Available: https://uyuni-project.org/

[13] “Red Hat Satellite.” [Online]. Available: https://www.redhat.com/en/technologies/management/satellite

[14] “ATIX orcharhino.” [Online]. Available: https://orcharhino.com/

[15] “SUSE Multi-Linux Manager.” [Online]. Available: https://www.suse.com/products/multi-linux-manager/