The SELinux problem that cast months-long shadow

A story about SELinux and tech debt.

Matyas Horky <mhorky@redhat.com>
Product Owner for Content Management and Data Collection,
Red Hat Subscription Management/Red Hat Insights

Red Hat

Disclaimer

- The content of these slides is my recollection and interpretation of the events.
- Nothing contained in this talk should be taken as an official communication of Red Hat itself.
- Please refer to the official documentation, support articles or open a customer case if you believe you've been affected and need more information.

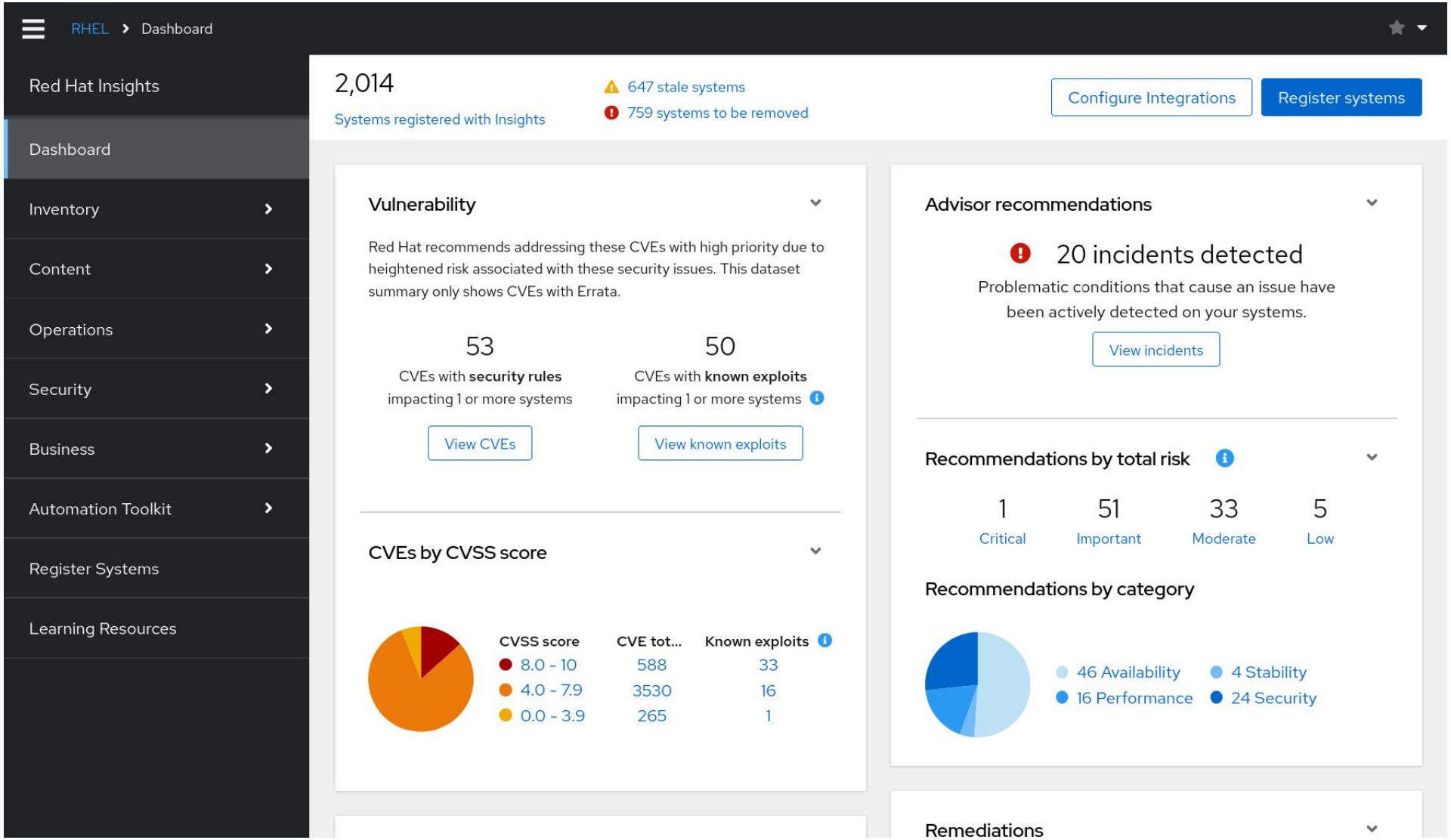
Who's talking?

- I am a Software Engineer and a Product Owner for Content Management and Data Collection.
- I'm part of the Client Tools team. RHEL, Red Hat Insights, Red Hat Satellite all meet here. Indirectly, we support products that integrate with RHEL, such as OpenShift or Podman Desktop.
- Our team doesn't have an upstream, we target RHEL while supporting Fedora-based distributions indirectly.

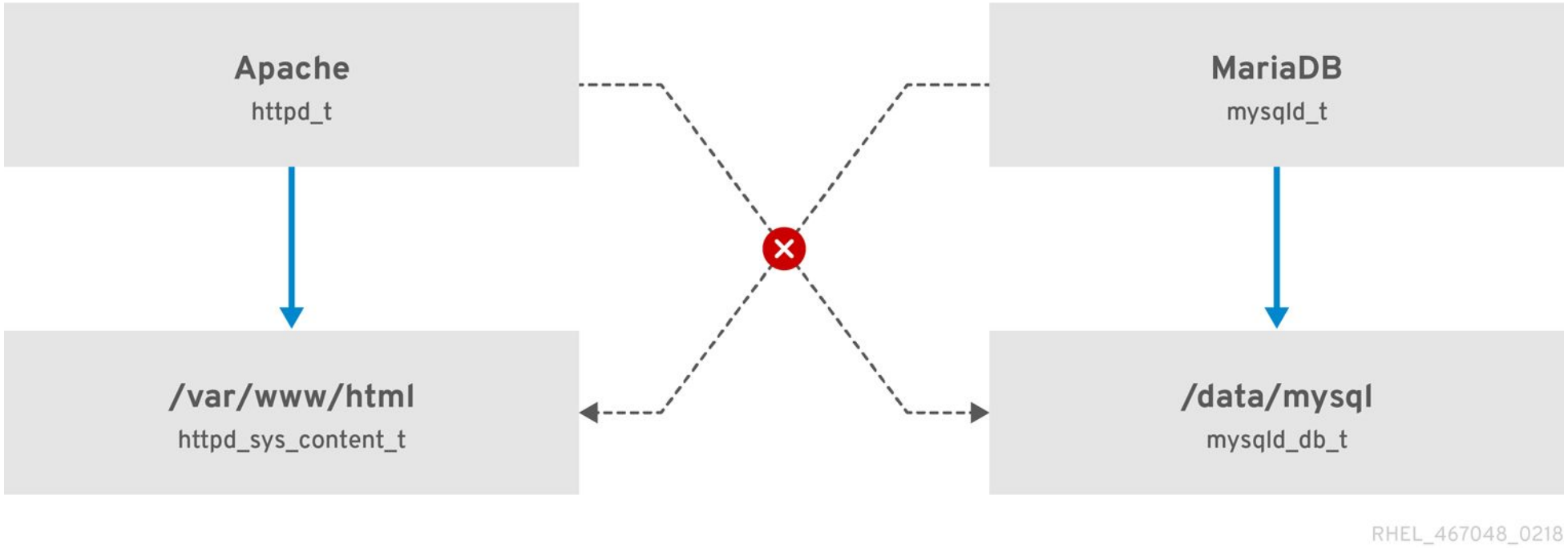
Keywords

What's the tech?

- Red Hat Insights, a software-as-a-service hosted by Red Hat.
- SELinux, a security architecture for Linux, part of RHEL.
- insights-client, a tool that collects data of RHEL hosts and uploads it to Red Hat Insights.



https://console.redhat.com/insights



https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/10-beta/html/using_selinux/getting-started-with-selinux

Intro

We had a history with SELinux

- 2022-05: Issue is filed by a support engineer:
  “insights-client is producing lots of AVC”.
- 2022-08: SELinux ships updated policy in RHEL 8.6+ & 9.0.
- 2022-11: Package changes policy again due to problems caused by the complexity of behavior.

/root/.gnupg annoyance

- 2022-04: “insight-clients tries to create /root/.gnupg”.
- 2023-09: The card is picked up by a developer.
- 2023-10: Code review starts.
  - One of the focus areas is GPG – its daemon may cause weird race conditions when it cleans up after itself.
- 2024-02: Quality engineer performs manual verification, the PR is merged.

Work done

- Mon 2024-02-12: The big insights-client outage starts. Engineering will not know of any problems for another week.

Actually, wait. How does that happen? We're talking about RHEL, doesn't it have release cycle, gating, component tests?

We'll need to take a small detour for a few minutes.

Client and Core

Client and Core

- insights-client repository is mostly an RPM shell around insights-core.
- Client is a minimal program that ships the configuration file, systemd services, and ensures Core is up to date.
- Core is a GPG-signed archive that contains collectors and parsers that Red Hat Insights use to display recommendations or ensure compliance.

Let’s look at the problem

The .patch

"We shouldn't touch directories in /root/."

- Abstract GPG away from the rest of the business logic.
- Instead of subshelling with no additional setup, create a temporary directory, use it as $GNUPGHOME and perform all actions in it.
- Once finished, clean up the temporary directory.

206 lines of Python, plus unit tests.

```
home = make-temporary-directory()
for key in keys:
    gpg --home home --import key
gpg --home home --verify signature file
del home
```


The problem

- Process gpg_t used to use ~/.gnupg with label gpg_secret_t.
- Python's tempfile.mkdtemp() uses /tmp/ by default, and created directories are labeled as insights_client_tmp_t.
- gpg_t isn't allowed to write to insights_client_tmp_t.

```
avc: denied { write } for comm="gpg"
 scontext=system_u:system_r:gpg_t:s0
 tcontext=system_u:object_r:insights_client_tmp_t:s0
 name="tmp4o9mzj2x" tclass=dir permissive=0
```


sudo grep denied /var/log/audit/audit.log

The result

- AVC denial results in OSError being raised.
- Unhandled error is propagated from Core into Client wrapper.
- Wrapper exits with non-zero status code as well.
- The systemd service transitions to FAILED.

So the service crashed, why is it bad?

- For historical reasons (read: tech debt), the service was configured as Type=simple, Restart=no.
- It's been like that since early 2018/RHEL 7.
  - RHEL 6 used cron, which runs every time no matter its history, and this behavior was replicated.
- It has been known that network/system glitch may cause the service to fail. It was never addressed; other issues more visible to the users were prioritized.

The systems are not reporting or self-updating

We actually have a problem

- Mon 12th: Core is published to production.
- Tue 13th: Server-side team discovers their component is broken due to dependency on insights-client.
- Wed 14th: Fix is released to production.
- Mon 19th: The bug is discovered by more engineering groups.
- Tue 20th: First emergency meeting is held.
  - Issue is reproduced. Support article is published.

Emergency meetings

- Who is affected?
  - RHEL 8.6+ and RHEL 9.0+ with SELinux enabled.
- How will Insights services react when the hosts stop reporting?
  - The hosts will get deleted along with their configurations.
  - Let’s pause the deletion and worry about it later.
- How many customers and systems?
  - How do we even figure this out?

- Client engineering doesn’t handle services, they don’t really know how many systems there are.
  - How to get the numbers?
  - How to even figure out who the relevant teams are?
- There is no precedent on informing customers about problems. What’s the best way to send the emails?

For three weeks, representatives from engineering, product management, support, and people management meet on a daily basis to prepare the next steps.

All the time, the discussions are constructive and everyone owns the problem. No one is blamed for the problems.

Why wasn’t it caught?

- Manual QE verification didn’t check for SELinux alerts.
- Automated QE tests didn’t check for SELinux alerts.
- Staging environments didn’t use staging Core.
- Service engineers don’t check their end-to-end pipelines on weekends.

We almost found it

- Similar improvement was also contained in the Client wrapper. The same week Core patch was merged, AVC denial is raised while making a release build.
- This issue is triaged next Monday, and a decision is made to revert both patches before the team investigates further.
- Bad Core was already in production by that point.

What have we learned?

- A lot.

Have emergency strategy planned

- Nothing is “too big to fall”, it’s all built from individual parts that can fail.
- It is important to make informed decisions. If you don’t have the underlying data, you’ll have to guess and hope.
- Be aware of what you are special in, you have to own both advantages and disadvantages.
  - Red Hat isn’t SaaS-first company, Insights is an outlier.

Responsible software development is about people

- Talk to your quality engineer counterparts, ensure they understand the problem. Don’t skip them, they guarantee the software works, you are creating problems for them.
- Talk to others. Extend visibility, establish contacts, make friends.

Communication outwards

- It’s far better to overcommunicate.
  - Knowing about issue I’m not affected by is far better than not knowing about a problem I am affected by.
  - Email filters exist if you get too annoying.
- It is hard to tell paying customers they are not getting the features yet. It is much harder to tell them they have to fix problems you caused.

Accept SELinux

- Yes, it is hard to get into.
- If SELinux is supported use-case, develop and test with it. That's the only responsible way of dealing with it.
- Test for AVC denials automatically. If it's done by a human, something will slip through.

Code reviews

- Everyone likes to focus on code style or how methods are split.
- Performing thorough behavioral verification is much harder, but much more rewarding when it discovers problems.
- The developer should put in QA hat as well, dedicated quality engineers will by definition never discover all edge cases contained in the source code.

Outro

We corrected a lot since then

- The .service file fix landed in RHEL 10 Beta and RHEL 9.5.
- We have a new maintainable integration test suite that catches regressions. Automatic SELinux checks are in early development.
- Engineering has access to usage trends now, giving us the possibility to see suspicious swings.

So how did the final fix look like?

```
home = make-temporary-directory(gpg-can-write-here)
for key in keys:
    gpg --home home --import key
gpg --home home --verify signature file
del home
```


Why did I tell the story?

- I am the engineer who authored the patch.
- I am the tech lead for the RHEL component.
  - I took over the role one week before these problems started.
- I am the product owner for the whole area.

Thank you

Any questions?

Matyas Horky
mhorky@redhat.com

Public-facing support article for the issue:
https://access.redhat.com/solutions/7056526

Red Hat