INTIGRITI

Scaling Vulnerability Management: Scale-Up to Enterprise

Niels Hofmans

Head of Security & IT at Intigriti

Freelance Cybersecurity Expert

Open Sourcerer

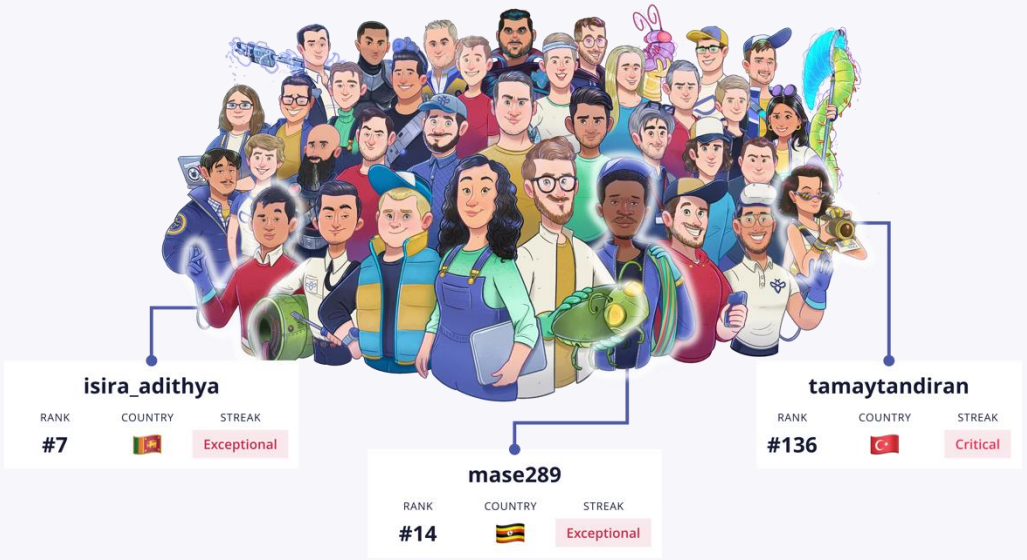
niels@intigriti.com

linkedin.com/in/nielshofmans/

hazcod.com

About Intigriti

www.intigriti.com



- Bug bounty platform in Europe
- Founded in 2016
- Headcount of 130
- SaaS & services hybrid
- Store confidential data as a service

OUR CUSTOMERS INCLUDE

Microsoft intel UBISOFT Coca-Cola Red Bull Nestlé VISMA Personio

A little history

INTIGRITI

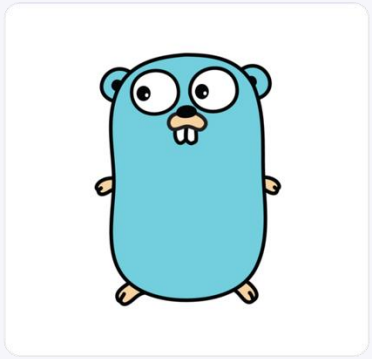
Phase 1

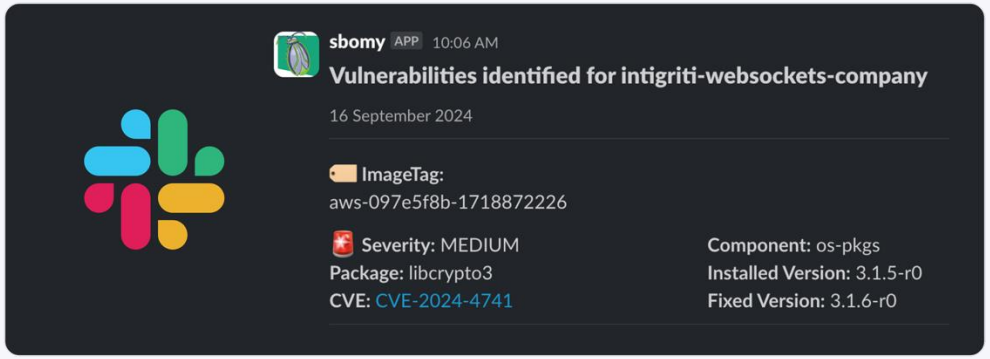
Manual labor

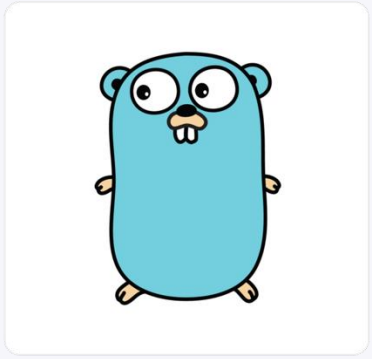
Source	Vuln. type	Cadence	Follow up
MDM	Device posture	1+x month	Manual via email
EDR	EDR Detections Software vulns.	1+x week 12+x week	Manual via Slack
AppSec/Bug bounty	Product findings	1+x week	Manual via Slack
News feeds	Trending CVEs	Avg. 769 CVEs /week	Manual via RSS

Phase 2

Spam automation

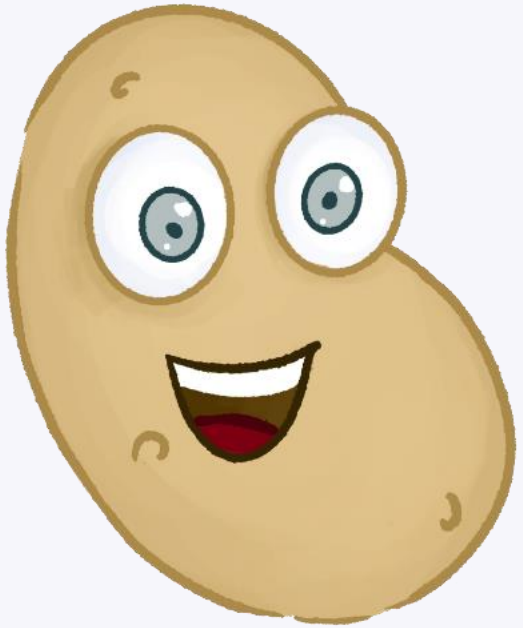






Issue

Potato overload

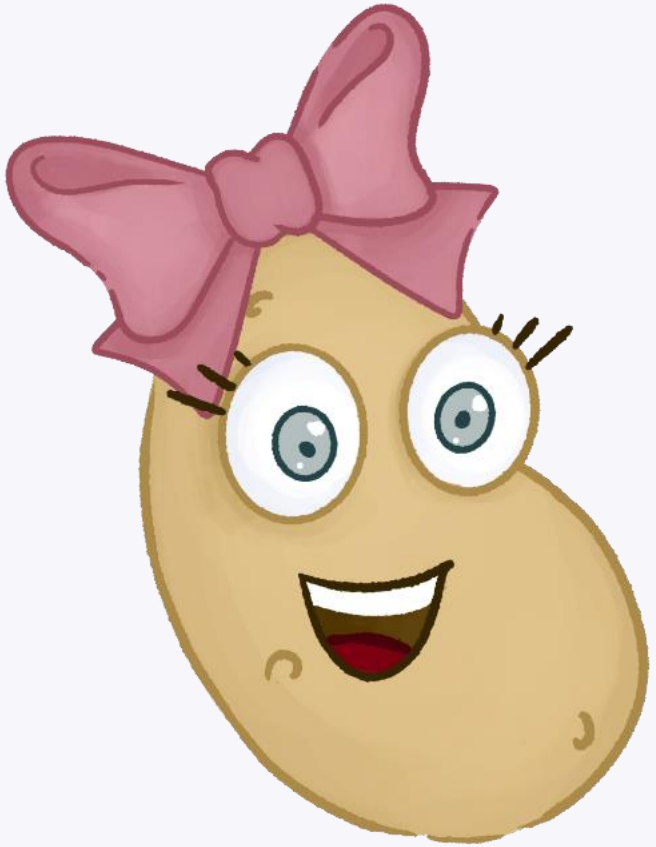


- We basically spammed others instead of ourselves
- Trying to patch everything doesn't scale
- This wasn't self-service

- DIY CRUD app was boresome
- Commercial solution was not ideal:
  - Costly
  - Theoretically we had all vuln data already

Idea

Potato overload



But… we had a (cloud) SIEM already!

- Re-use data integrations and SDK knowledge
- Log-based meant historic views
- No painful storage or ACL woes
- Deploy via CI/CD
- Enrich vuln. context from providers

Issue

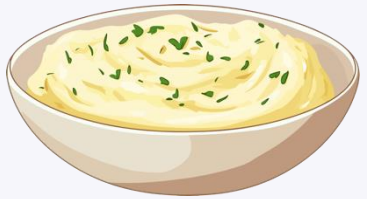
Potato overload

Although...

- We'd abuse a log system as a relational database
- We'd have to maintain critical infrastructure
- Yet another data schema? xkcd/927

Let's just do this.

Project Mashed Potato



MDM
+
EDR
+
AppSec
+
CSPM & Containers
+
TI
+
Internal testing

+

Normalization

+

Storage

SIEM

+

Dashboarding

Phase 3

Plenty of normalized vuln. data

AssetCriticality	Unassigned
CveId	CVE-2024-53076
Description	In the Linux kernel, the following vulnerability has been resolved: iio: gts-helper: Fix memory leaks for the error path of iio_gts_build_avail_scale_table() If per_time_scales[i] or per_time_gains[i] kcalloc fails in
ExploitStatus	0
HostLastSeen [UTC]	2025-03-06T00:00:00Z
IpAddress	192.168.198.130
ProductName	linux-signed-hwe-6.8
ProductNameVersion	linux-signed-hwe-6.8 6.8.0-52.53~22.04.1
Reference	http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2024-53076
> Remediations	[{"type":"PATCH","action":"No fix or vendor remediation available for linux-signed-hwe-6.8 on Ubuntu 22.04. Alternatives might include software removal, OS upgrade, or other mitigating controls"}]
Severity	medium
TimeGenerated [UTC]	2025-03-19T15:02:11Z
UserEmail	[redacted]@intigriti.com
VulnerabilityType	Endpoint Vulnerability
> Affects	{"HostId":"5206341552d64b[redacted]","HostName":"[redacted]","Type":"CrowdStrike Sensor"}
TenantId	03914fb4-fb1c-4e3a-8b21-c03ea31a7453
Type	VulnerabilityManagementLogs_CL

Phase 3

Pretty SIEM dashboards!

- User wall of shame (“projects”)
- Top 10 vulnerable hosts
- Public CVE exploits
- Actively exploited CVEs
- Most vulnerable products
- Product backlog
- …

Phase 3

Augmentation!

Inject context into vulns.

- Add business **metadata** to assets (prod. vs. test, triage laptops, ...)
- **Ignore** or lower certain classes
- Link **Threat Intelligence** telemetry
- Re-calculate **likelihood** of exploitability

-> Recalculate severity

Phase 3

Workings

1

Dashboard per stakeholder

2

Daily follow-up of trends

3

Weekly consults with stakeholders

4

Manual querying for TI-based intel

5

Focus on risk, not numbers

Phase 3

First-world problems

So we built dashboards for stakeholders. But...

1

Too much data to chew on

2

Part of the data was always irrelevant

(auto patched)

3

Still 40K vulnerabilities

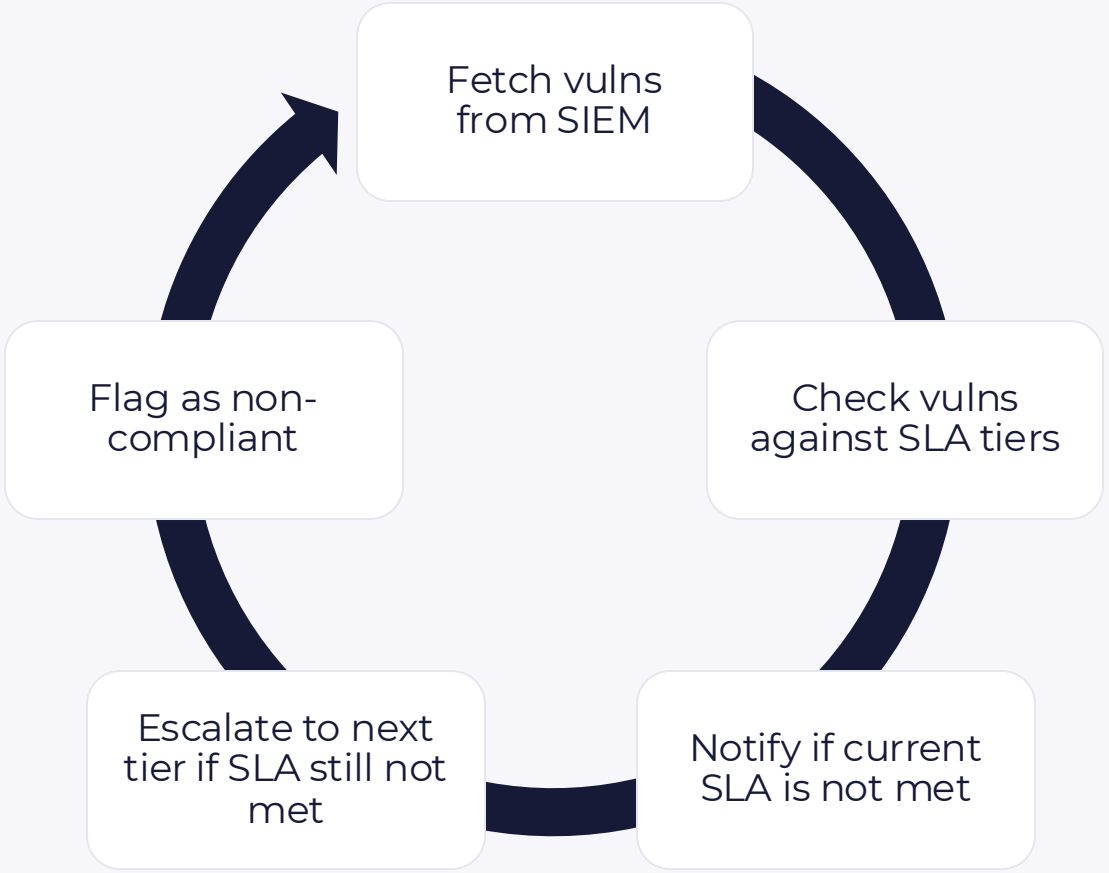
(non-duplicated)

4

Further querying was hard to sell to stakeholders

Phase 4

~~AI~~ Moving window prioritization




```yaml
notifiers:
  slack:
    webhook: https://slack.com/webhook

  email:
    sender: vulnmanagement@intigriti.com


filters:
  -
    severity: critical
    notifier: slack
    sla: 0h
    properties:
      exploited: true
      production: true
  -
    severity: critical
    notifier: email
    sla: 0h
    properties:
      exploited: true
      production: false
  -
    severity: ["High", "Critical"]
    notifier: slack
    sla: 8h
    properties:
      production: true
```

Phase 4

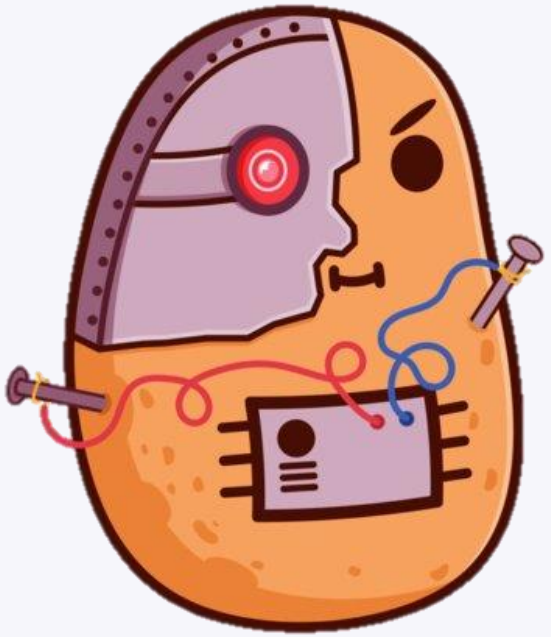
Potato stew

Handle vulns as cattle, not pets

- Give regular processes **time** to resolve (“patched by design”)
- Only **Slack** alerts for things that matter
- Only escalate to compliance when necessary
- Revisit filters every week

Phase 5

TBC: Cyber Potato

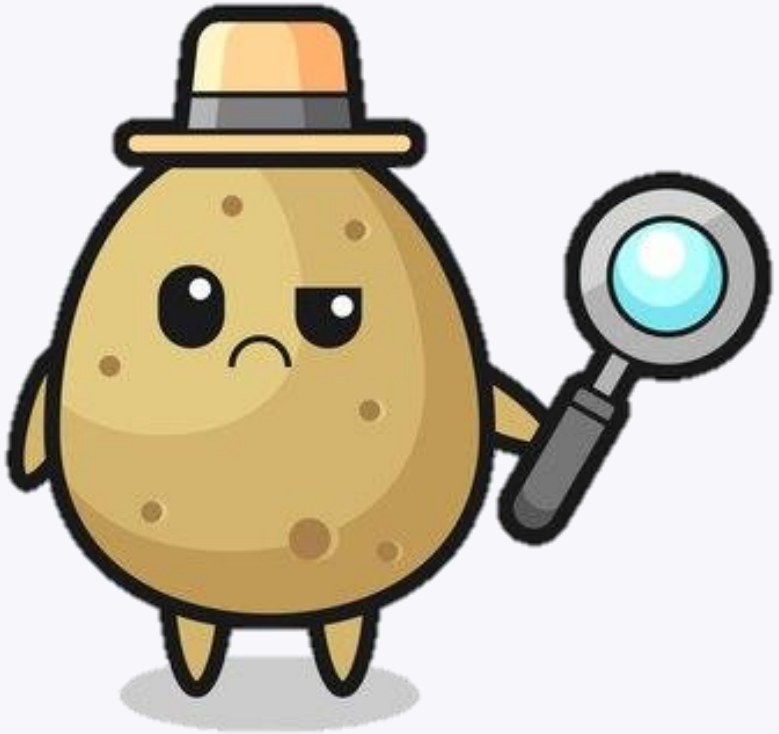


- Custom Slack App for notifying
  + add business logic; **Mute, Escalate**
  e.g. Mute for X days, Mute for all linux hosts, …

- Push custom device/user compliance logic to Entra ID
  e.g. no access to prod if;
  - Public exploit
  - With mitigation
  - Confirmed by security team
  - Alert via Slack without confirmation under SLA

Tater hunt

- There are two potatoes **hidden** across the venue
- Bring them **back** to their rightful place – the Intigriti patch
- Get free **tickets** to the Tuesday **afterparty**



INTIGRITI

Thank you

Questions?