Detection Engineering: Developing and Optimizing Attack Detection

Originally, this presentation was in German. AI helped to translate and optimize this presentation. In addition, it was used to generate graphics. See references. The access to all resources took place in August/September 2026.

Dr. Nils Timotheus Kannengießer

Agenda

- Status quo - Current threats in Germany in 2026
- What is Detection Engineering?
  - Typical threats
  - Examples of logs and their formats
  - SIEM systems, their agents and data protection
    - Decoders and rules
  - Optimized attack detection, examples & demo overview
- Conclusion

Status quo - Threats in Germany in 2026

Attacks by foreign intelligence services increased:

2023: 7% ==> 2025: 28% ==> 2026: 37%

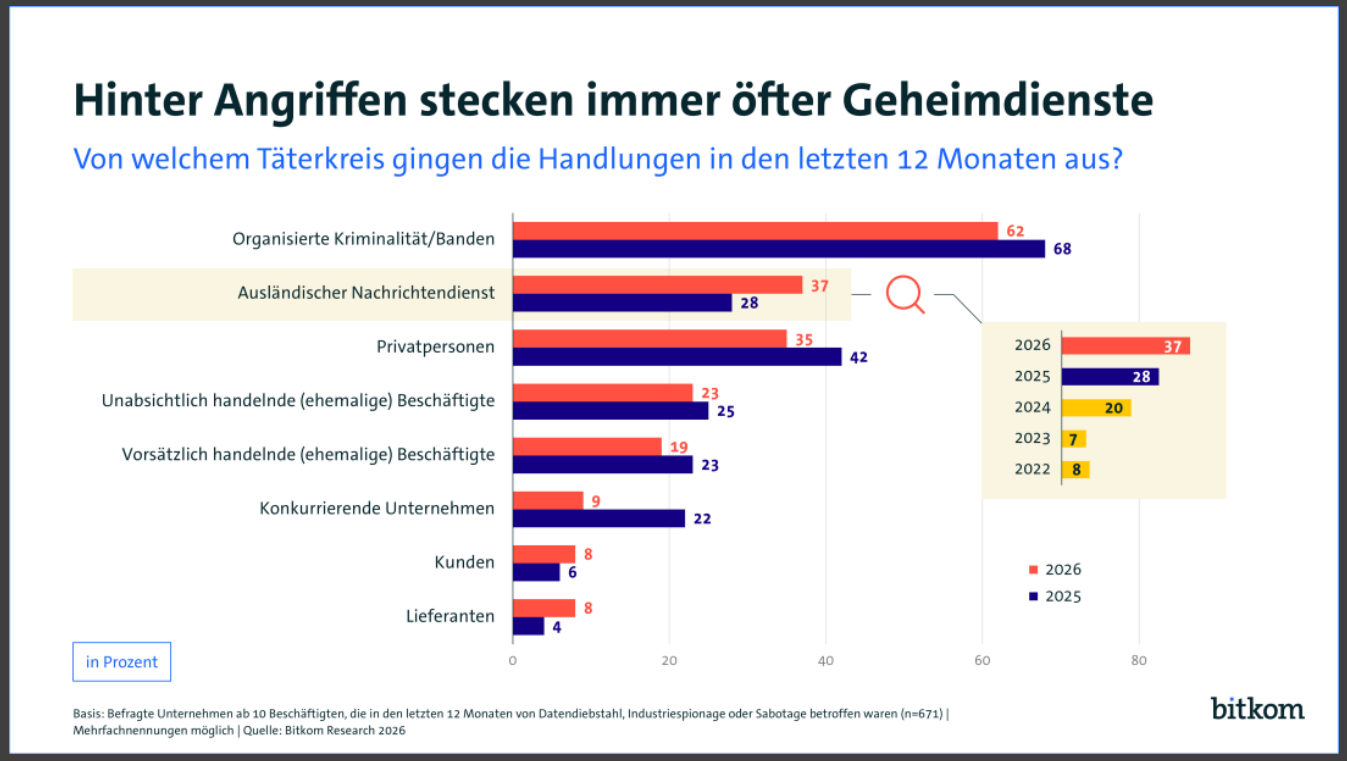


Major threats from China/Russia ... 96% of about 1,000 surveyed German companies reported attacks (67% confirmed + 29% suspected)

Source (including graphic): BITKOM - https://www.bitkom.org/Presse/Presseinformation/Angriffe-auf-deutsche-Wirtschaft-Spur-auslaendische-Geheimdienste

Status quo - Threats in Germany in 2026

Alongside highly specialized intelligence services, all other actors decreased. A growing number of less skilled attackers benefit from AI.



"The key is to detect and repel attacks early" (translated)

Bitkom President Dr. Ralf Wintergerst

Source (including graphic): BITKOM - https://www.bitkom.org/Presse/Presseinformation/Angriffe-auf-deutsche-Wirtschaft-Spur-auslaendische-Geheimdienste

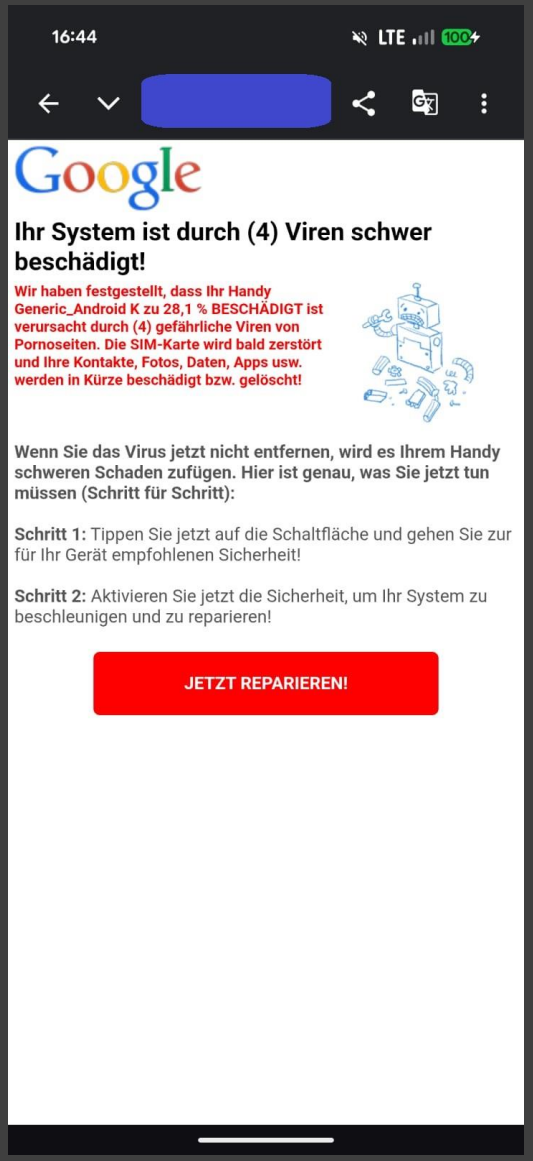
Status quo - Threats in Germany in 2026

In addition, private users are targeted every day as well: by clicking email links (shown here) or opening (aggressive) in-app ads.

Who has seen something similar in the past month?



What can we do about it... ?



Detection Engineering - Definition

Detection Engineering is a continuous, threat-informed cycle ("Direction", e.g. CTI), gathering relevant data sources ("Collection", e.g. logs), plus data preparation, analysis and rule development (e.g., SIEM, YARA, Sigma) in detail ("Processing") ...

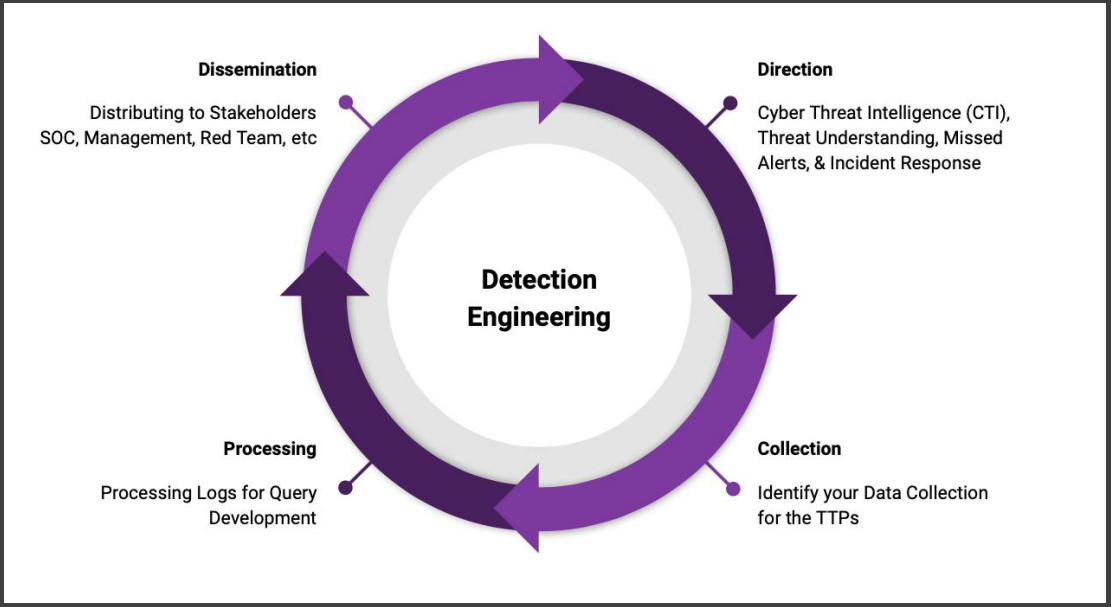


Illustration by SANS Institute (source below)

... which includes testing followed by the actual handover to operational SOC teams ("Dissemination"). Their feedback (false positives, insights from alerts, etc.) feeds back into the cycle.

Source (including graphic): SANS Institute - https://www.sans.org/blog/purple-teaming-threat-informed-detection-engineering

Detection Engineering - Threats

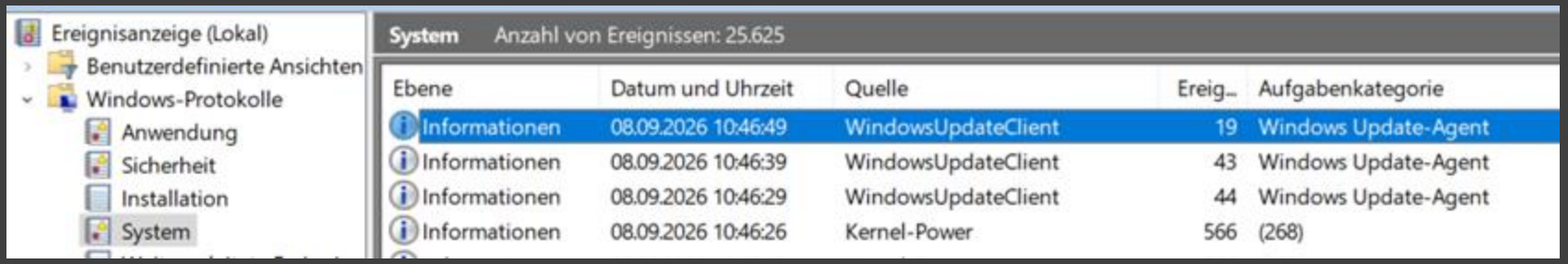
Summary:
Systematically developing, testing, operating and optimizing detection rules for security-relevant events.

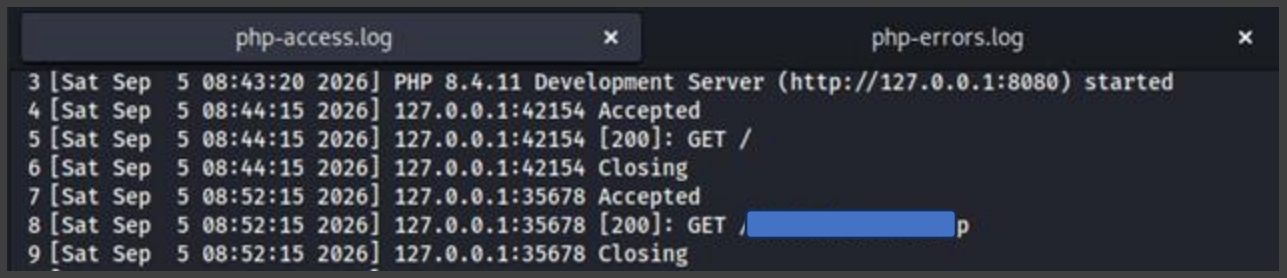
Typical threats and their telemetry (examples)

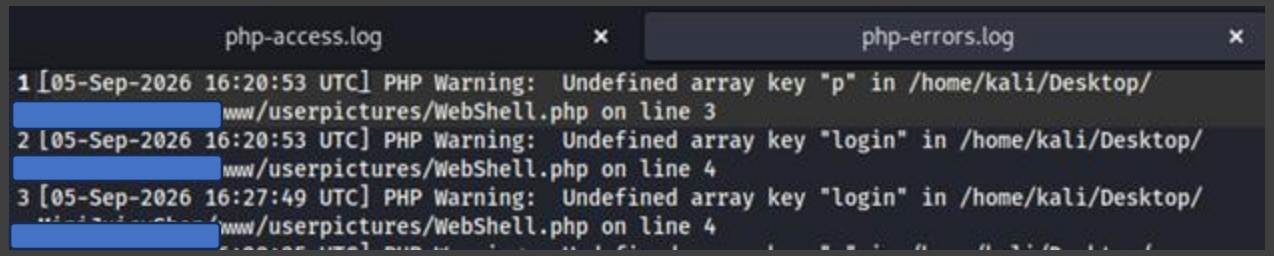
- Bot attacks and other attacks on web servers
  => Web server logs, new/modified files, ...
- Phishing emails
  => Logs, senders, attachments, logs of visited URLs, ...
- Malware execution
  => Suspicious processes/registry/files, traffic logs
- Data exfiltration
  => Suspiciously large transfers in real-time analysis, ...

Detection Engineering - Logs and Formats

How do typical logs look like? What formats exist? (Examples)







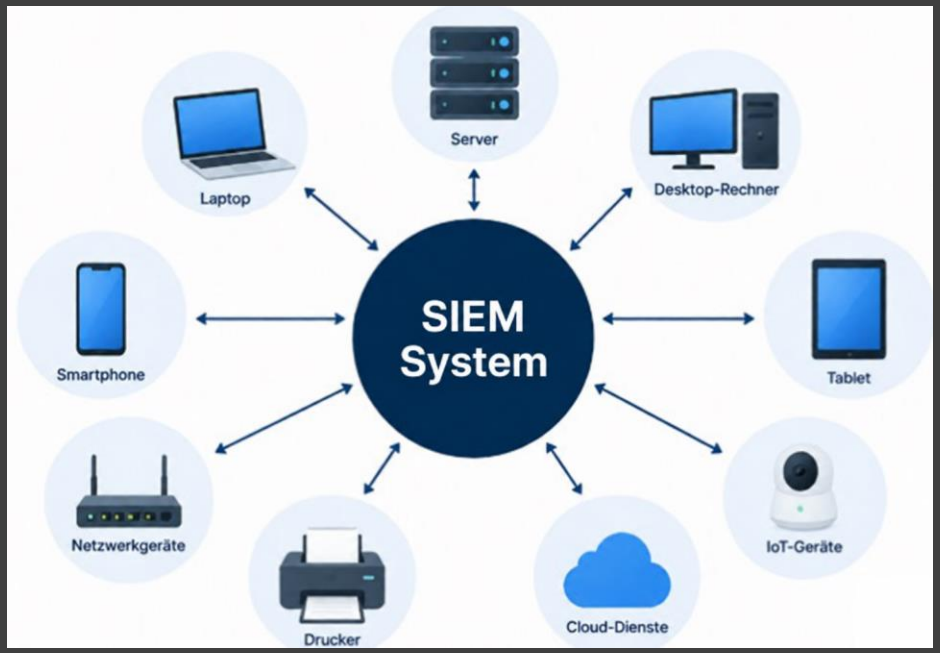
Type, date, time, source, event, additional information, service, ...

Detection Engineering - Logs, Formats and Agents

Logs may use text-based or binary formats.

For (detailed) logs, GDPR implications needs to be observed (e.g., personal use of a company phone or laptop).

Image source: generated on request by ChatGPT

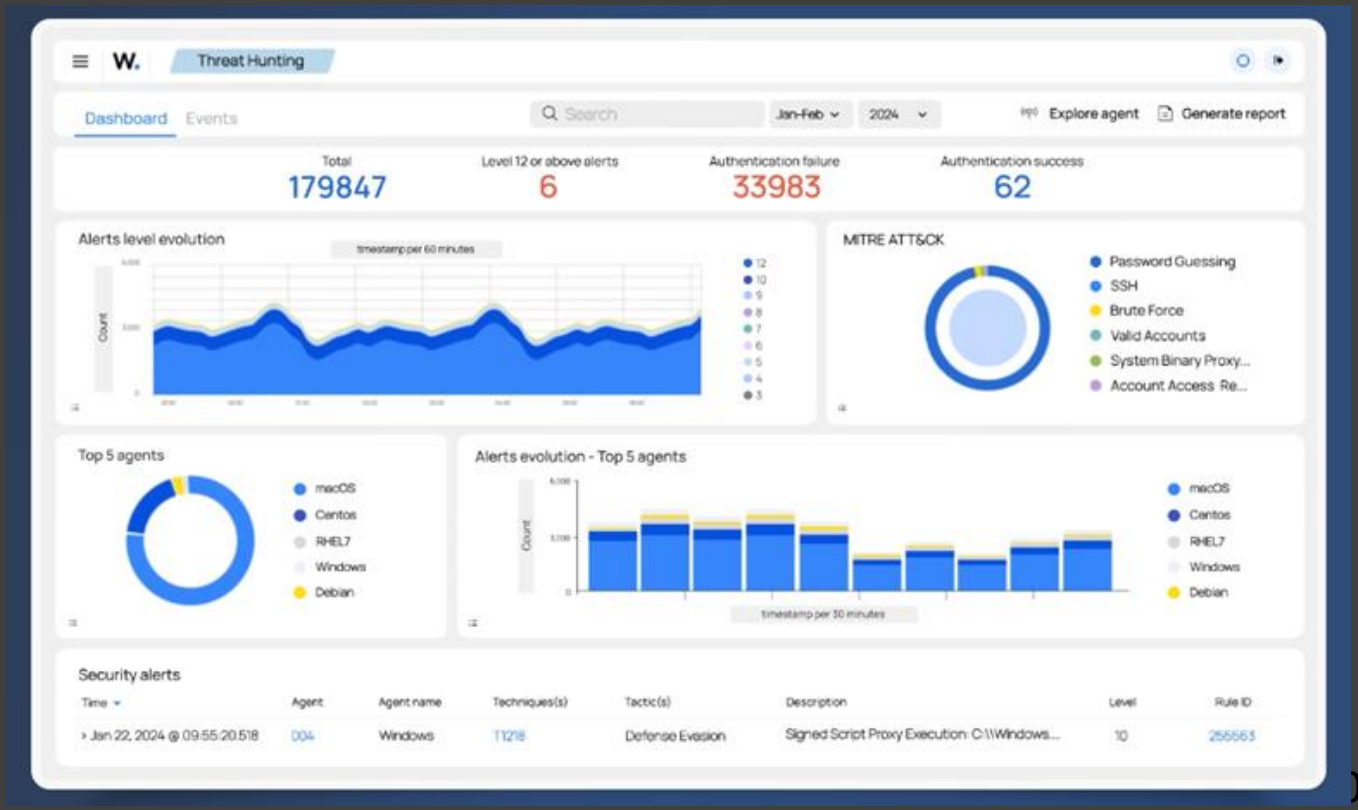


So-called agents send relevant data (possibly a selection of logs already) from each device to a central SIEM system.

Detection Engineering – SIEM (Security Inform. & Event Management)

SIEM solutions range from commercial to open source tools, e.g., Microsoft Sentinel, Splunk or **Wazuh (today's example)**.

They support (real-time) monitoring and attack detection, as well as management, MITRE ATT&CK classification and related tasks. See screenshot by manufacturer.



Graphic source:
WAZUH -
https://wazuh.com

Detection Engineering - "RegEx"

SIEM systems (MS Sentinel, Wazuh, Splunk) process structured events and extract information from unstructured logs using parsers and regular expressions.

Example: ^ 5 \d \d $

^ means line start outside [ ], otherwise negation ($ means line end)

\[matches [

[^\]]+ matches one or more characters other than]

[abc] matches one of these letters

[^abc] matches none of these letters (e.g. another letter or digit)

\[[^\]]+\] matches [anything except]]

\s whitespace character (e.g., space)

\s+ one or more whitespace characters

\s* zero or more whitespace characters

\S+ one or more non-whitespace characters

\d+ one or more digits

\S+:\d+ e.g. IP address : port

\d{3} exactly three digits

\d{2,4} between 2 and 4 digits

. any character

[A-Z]+ one or more uppercase letters

( ) capture for output

(?i) ignore case

Detection Engineering - Decoders for a SIEM System

To interpret different logs correctly, so-called decoders are needed (valid entry? Field mapping possible?)

Example decoder (Wazuh, incomplete reg.exp.)

[Sun Sep 12 11:20:03 2026] 127.0.0.1:46504 [200]: GET /index.php

```
<decoder name="php-demo-access">
 <prematch type="pcre2">
 \[[^\]]+\]\s+\S+:  ...  i.e. [Sun Sep 12 11:20:03 2026] 127.0.0.1:      (excerpt)
 </prematch>

 <regex type="pcre2">  ... </regex>

 <order>php_timestamp,srcip,srcport,http_status,http_method,url</order>

</decoder>
```

Notice: Wazuh field Dynamic (custom) field

Detection Engineering - Rules for a SIEM System

... while defined rules detect specific events and assign their severity.

Example rule definition (Wazuh)

```
<rule id="100102" level="7">
   <if_sid>100100</if_sid>
   <field name="http_status" type="pcre2">^5\d\d$</field>
   <description>PHP demo server returned HTTP $(http_status) for $(url)</description>
</rule>
```

^5\d\d$ matches 500-599 condition

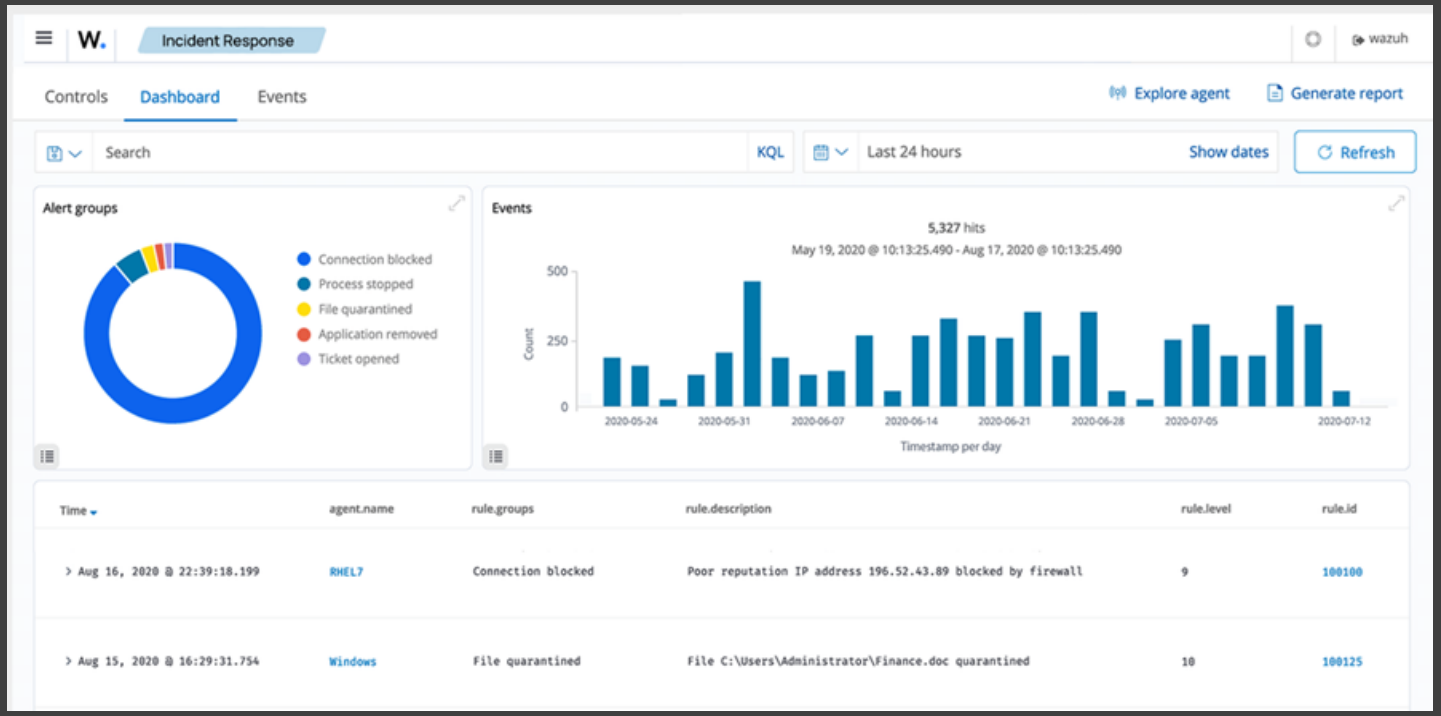
Custom, unique ID (Wazuh convention: 100000-120000)

Custom level from 0 (irrelevant) to 15 (severe)

Source: https://documentation.wazuh.com/current/user-manual/ruleset/rules/rules-classification.html

Detection Engineering - SIEM Events and Automation

There are many different events, from information ("Server started") to security-relevant events ("Trojan XY found"). Predefined rules/levels simplify administration and enable automatic responses (e.g., "file quarantined"/see screenshot by manufacturer).



Graphic source: WAZUH - https://wazuh.com/platform/xdr/

Event = observed occurrence

Alert = triggered detection

Incident = assessed security incident

Detection Engineering - Optimization

Rules can be revised as needed, in line with the Detection Engineering cycle, but...

Do you think using AI (instead of rules) would be much easier?
"Please assess this attack (raw data XY) and provide an executable action (for Wazuh, firewall, etc.)." => OpenAI provides SRE agents (see below)



Image source: generated by ChatGPT on request

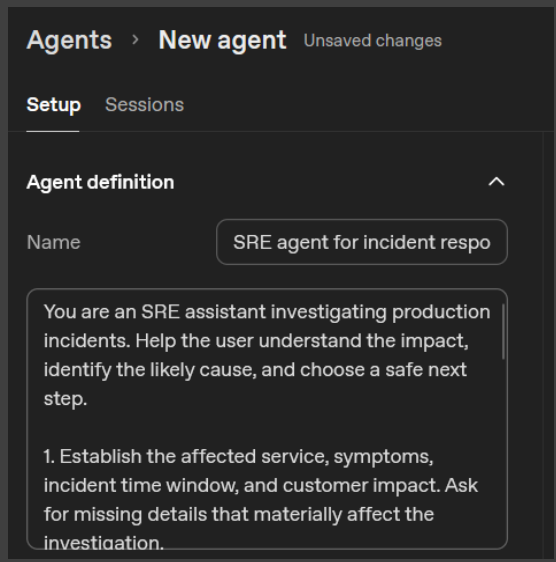


Image source: ChatGPT.com Agent definition



Would this be a good option?

Ask AI to analyze everything? Are there any risks?

Detection Engineering - Optimization

- Probably not a good solution.
  - (GDPR: data requiring protection.)
  - Without a review, AI should (for now?) not make decisions on its own: risk of incorrect decisions and attacks (prompt injections). Possible legal restrictions (EU AI Act).
  - (Very high token costs for AI usage.)

However:

Individual tasks (e.g., preparing a ticket) can be passed to AI for recommended actions and prioritization (level). A human may validate the result (= reduction of workload).

Detection Engineering - Optimization

- Wazuh is testing various scenarios, e.g.,
  - Enhancing security audits: Nmap scan with detailled information on a found service
  - Simplified configuration and rule generation using an actual log entry to let AI create the required rule
  - Enriching YARA detection with more information (e.g., information about found Mirai malware)

Good to know:
YARA rules classify files using text or binary patterns, e.g., with publicly available signatures (Valhalla). Sigma is the same for logs.

Sources:
WAZUH - https://wazuh.com/blog/nmap-and-chatgpt-security-auditing/ and https://wazuh.com/blog/a-sneak-peak-at-agentic-ai-in-wazuh/ and https://documentation.wazuh.com/current/proof-of-concept-guide/leveraging-llms-for-alert-enrichment.html as well as NEXTRON SYSTEMS - https://valhalla.nextron-systems.com/

Detection Engineering - Optimization

Summary: AI can help to optimize existing detections.

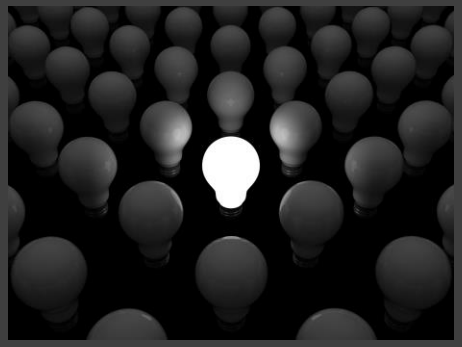
Ideas & examples (by me):

When APK file alerts occur, trigger AI-assisted reverse engineering of unknown apps of employee devices (consider data protection!) to obtain a detailed assessment (first impression: very good/ongoing research).

Correlate detected attacks using AI to identify and prevent imminent data exfiltration early.

[...] Any other ideas?





Detection Engineering - Optimization

A note on AI:

Many capabilities already exist or are under active development.

Mobius

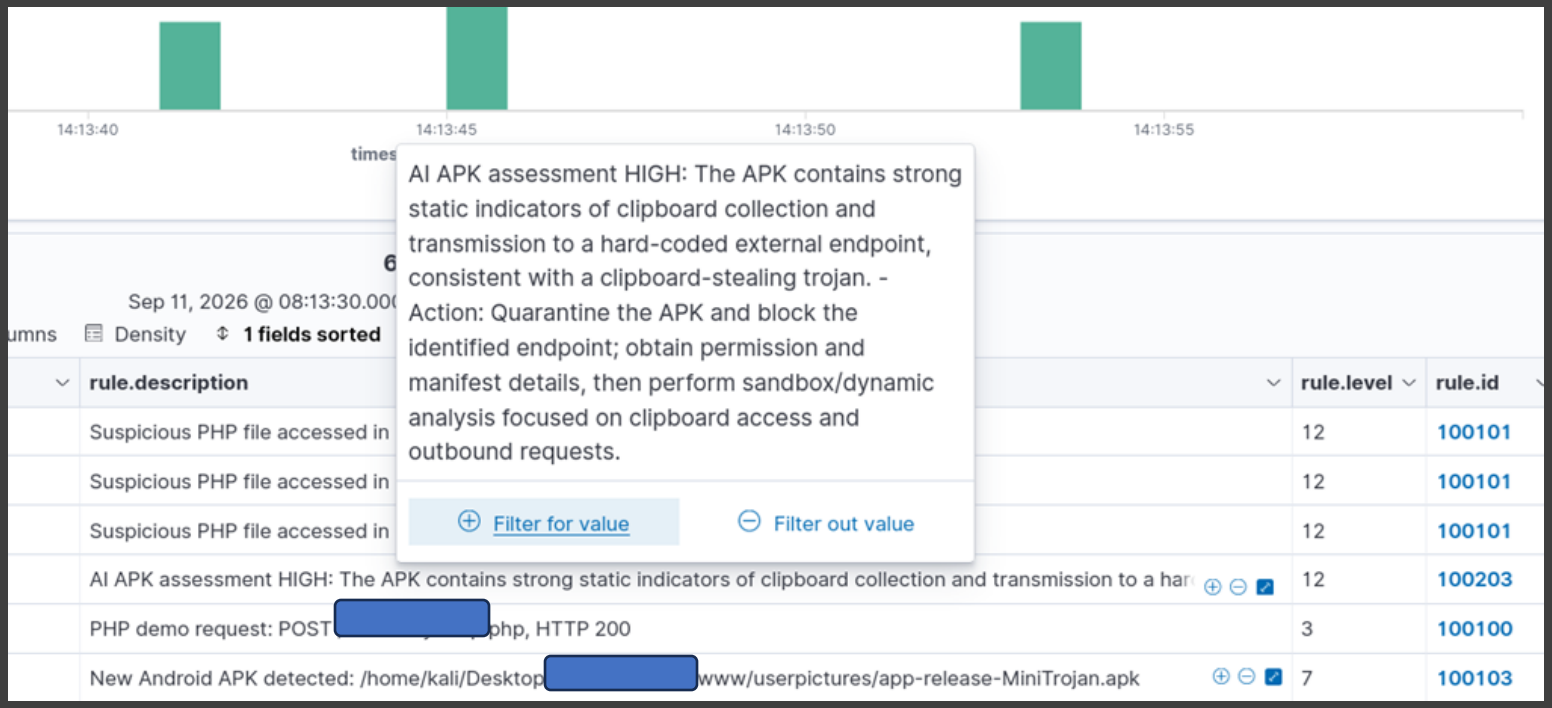
"Mobius is an agentic SOC for Wazuh. It ingests your alerts, triages every one of them with AI, enriches with MITRE ATT&CK, correlates the escalations into incidents, and can respond, automatically or with your approval."

… but: Do you want to export all alerts to an AI, maybe worse outside Europe? GDPR?

Source: WAZUH - https://docs.mobius.wazuh.com/

Detection Engineering – Demo overview

- Demo web server
- Scenario: Shop with insufficient upload validation
  - WebShell + replacement of mobile app with malicious one
- Wazuh with predefined monitoring (files, access) rules
- AI for further analysis (example: malicious Android APK)



Detection Engineering - Conclusion, Thanks & Discussion

- **Threat-informed development:**
  Attack scenario --> required logs --> detection rules

- **Systematic improvement:**
  Test rules and optimize them using SOC feedback

- **Targeted use of AI for optimization:**
  Rule development & alert assessment

Thank you for your attention

Questions? Discussion?

Dr. Nils T. Kannengießer