SOC detection engineering and management

This page deals with SOC detection engineering and management (detection use case creation, detection capabilities assessment, etc.)

ToC

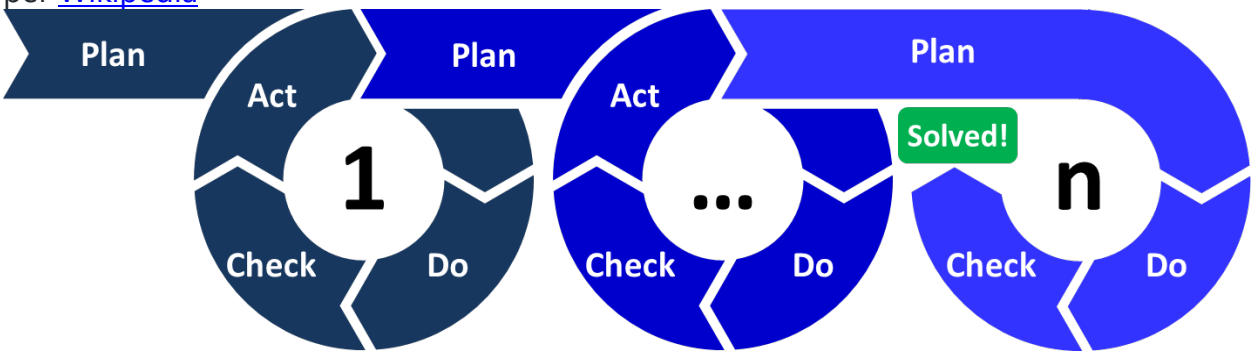
Must read

- MITRE, top TTP for ransomwares
- Yogosha, SIGMA Rules: how to standardize detections for any SIEM
- Ch33r10, Enterprise purple teaming
- F. Roth, Detection engineering cheat sheet
- SIEM rules publications:
  - Sigma HQ (detection rules)
  - Splunk Security content (free detection rules for Splunk)
  - Michel De Crevoisier's Git
- Known exploited vulnerabilities:
  - CISA catalog
- Medium, 'About detection engineering'
- NCSC, Detection Practices
- Intrinsec, 'Limitations of MITRE ATT&CK' (in FR)
- LinkedIn, Risk assessment with ISO 27005
- PECB, ISO 27001:2022, what are the changes?
- ANSSI, EBIOS RM methodology
- David J. Bianco, Pyramid of pain
- Atlassian, CI/CD/CD

Generic recommended approach

PDCA multi-loop

As
per [Wikipedia]



PDCA being applied to SOC

Plan

Sensors:

- Determine which sensors or endpoint/app logs, you miss in terms of detection capabilities.
- Determine how to generate and ingest their logs in the SIEM.
- Build a project agenda.

SIEM rules:

- Determine which detection logic you miss, directly in the SIEM.
- Build a project agenda (deployment).

Detection automation playbooks:

- Determine which automation you miss, based on feedback from previous alerts and incidents handling.
- Build a project agenda (deployment).

Response automation playbooks:

- Determine which automation you miss, based on feedback from previous alerts and incidents handling.
- Build a project agenda (deployment).

Documentation:

- Double check which Standard Operating Procedures (SOP), and global processes, you may miss or need to update.

Do

Sensors:

- Ingest the logs of the security sensor, or endpoint/app logs, that you previously identified.
- Make sure your data ingestion is compliant with the datamodel you use.

SIEM rules:

- Create the detection rules (SIEM searches) that match your previously identified needs.
- Create the alert objects in the SIEM or SIRP, to contain the contents of the SIEM searches in case something is found.

Detection automation playbooks:

- Implement the needed automation, first by drawing the process and procedures (my recommendation is to use BPMN);
- and then by implementing it in the SOA.

Response automation playbooks:

- Implement the needed automation, first by drawing the process and procedures (my recommendation is to use BPMN);
- and then by implementing it in the SOA.

Handling procedure (SOP):

- If it does not exist already, create the handling procedure for the newly created detection rule.
    - My recommendation: take the french CERT-SG IRM procedures as templates.

Check

Logs:

- Make sure your data ingestion is compliant with the datamodel you use (or, at least, the SIEM one).

Handling procedures (SOP):

- Make sure that the handling process and procedures are clear and working fine, for the tested alerts.

Automations:

- Make sure that the automations capabilities to help in the **detection** phase, work as expected (ie.: observables enrichment in the SIRP with queries to the TIP).
- Make sure that the automations capabilities to help in the **response** phase, work as expected (ie.: containment steps), by assessing it with purpleteaming.

SIEM rules [first run for the assessed detection capabilities]:

- Test the detection logics with narrowed use cases (specific events, that are generated on demand).

SIEM rules [following runs for the assessed detection capabilities]:

- Assess your detection capabilities with purpleteaming.
- Report your results and findings in a purpose-built app like Vectr.

SIEM objects:

- Assess the relevance and freshness of inclusion lists, aka whitelists (that are supposed to be synced with Git)

- Assess the relevance and freshness of exclusion lists, aka blacklists (that are supposed to be synced with Git)
- Assess the relevance and freshness of IOC lists (that are supposed to be synced with the TIP).
- Assess the relevance and freshness of assets lists (that are supposed to be synced with Git), for instance groups, VIP/VOP, particular endpoints, etc.

Act

- Fix everything that was previously identified as not working, missing, or not matching your needs.

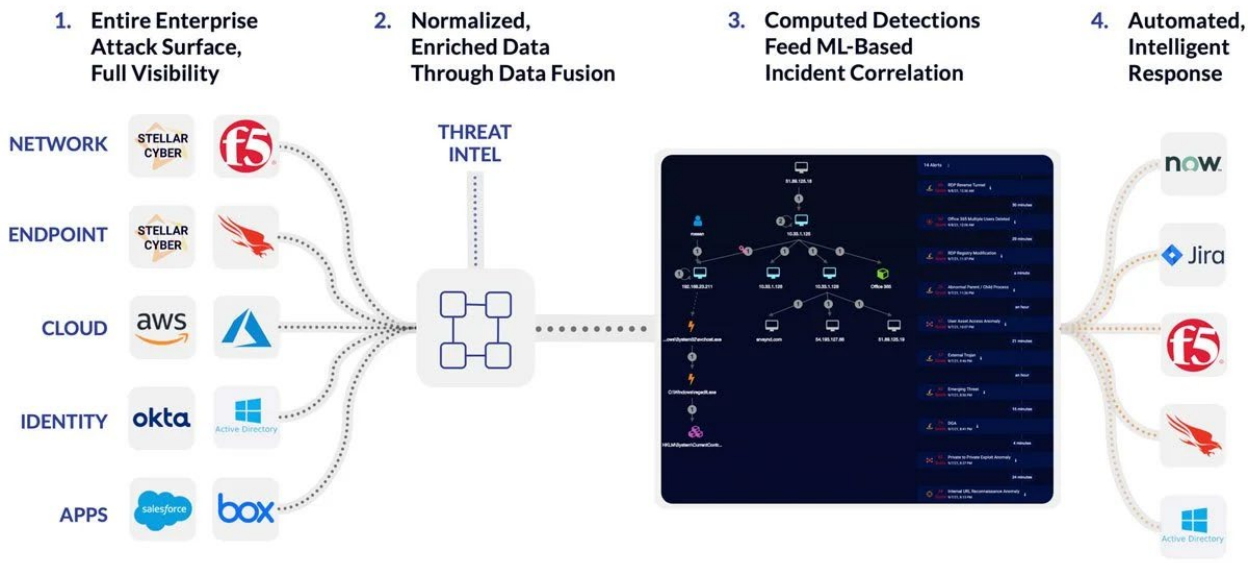
How to feed the "Plan" phase

Standard maturity and needs

Implement an XDR-like approach:

- Leverage threat intel;
- Leverage built-in detection capabilities of your security solutions/sensors;
- Implement correlations between those security sensors alerts, within the SIEM.

Here is a suggested global overview of it, through Open-XDR approach of Stellar Cyber:



TTP detection priorities identification:

- Use MITRE Engenuity calculator:
  - Calculate your top 10 TTP, based on your existing logging and detection capabilities.
  - Focus on the top TTP for ransomwares:
    - T1486: Data Encrypted for Impact, T1490: Inhibit System Recovery, T1027: Obfuscated Files or Information, T1047: Windows Management Instrumentation, T1036: Masquerading, T1059: Command and Scripting Interpreter, T1562: Impair Defenses, T1112: Modify Registry, T1204: User Execution, T1055: Process Injection.
- Leverage daily watch to maintain your knowledge about current most commonly used TTP:
  - for instance: Recorded Future 2021 top TTP report:
    - T1486 (Data Encrypted for Impact), T1082 (System Information Discovery), T1055 (Process Injection), T1027 (Obfuscated Files or Information), T1005 (Data from Local System).

Leverage the native detection coverage of IT environments:

- Refer to Security Stack Mappings
  - for AWS;
  - for Azure;
  - for GCP.

Leverage the documented detection coverage of security solutions:

- Refer to Security Stack Mappings
    - Regarding Vectra.

Cyber watch:

- SIEM rules publications to keep an eye on:
    - Sigma HQ (detection rules).
    - Splunk Security Essentials (free detection rules for Splunk).
    - Elastic rules.
    - Michel De Crevoisier's Git.
    - SOC Prime.
    - ThreatHunterPLaybook.
    - CAR, MITRE Cyber Analytics Repository.
    - Elastic prebuilt rules.

Focus on top relevant vulnerabilities:

- Vulnerabilities that are confirmed commonly exploited in the wild (see CISA KEV);

AND

- that are confirmed as valid (unpatched) within your organization.

Then try to implement detection rules that are specific to those carefully selected 0days.

My recommendation, still, is to make sure not to spend all your time running after latest 0days, as it is time consuming and not that efficient in the end in terms of working SOC detection capabilities.

Advanced maturity and needs

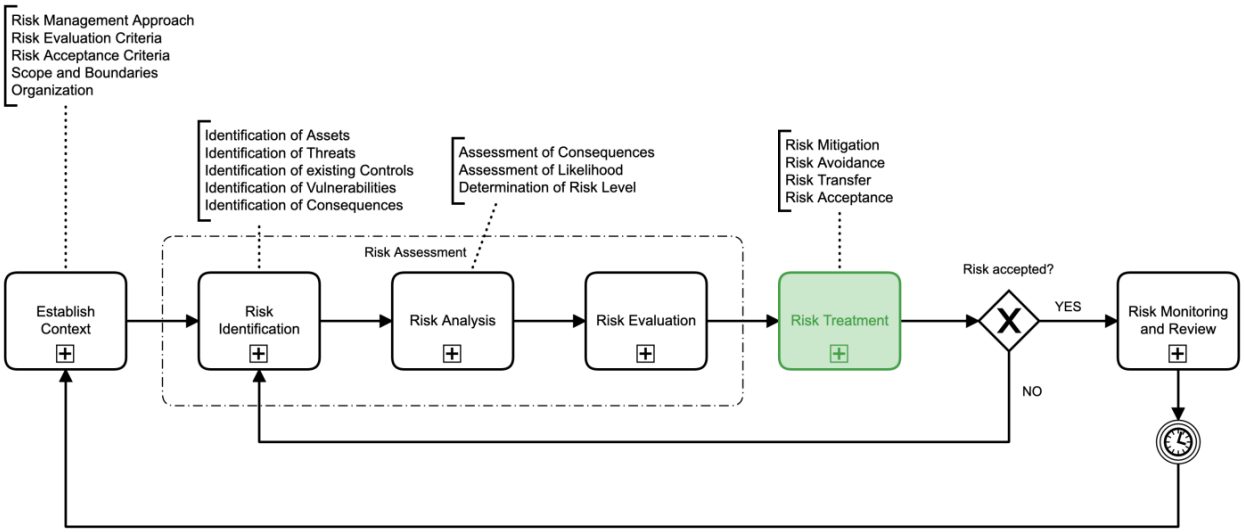
Precisely define your needs and the SOC priorities:

- Leverage a risk management-based approach, to determine:
    - threat actors (if possible);

- critical assets;
- attack scenarios (somewhat, kill chains and TTP).

Identity risks and their treatment:

- Here is a simplified drawing of the global process, leveraging ISO 27005 approach:



Define risk prioritization as per BIA:

- Here is a drawing of this approach, from NIST, leveraging a Business Impact Analysis to determine risk prioritization (from NIST IR 8286A, see To Go Further / Must Read):

Fig. 4. Elements of Information Risk Identification (from NIST IR 8286A)

The completion of the risk description column is composed of four activities that are detailed in NIST IR 8286A, Subsections 2.2.1 through 2.2.4. The activities include: • Part A – Identification of the organization's relevant assets and their valuation • Part B – Determination of potential intentional/unintentional threats that might jeopardize the

confidentiality, integrity, and availability of those assets • Part C – Consideration of vulnerabilities or other predisposing conditions of assets that make a threat event possible • Part D – High-level evaluation of the potential consequences if the threat source (part B) exploits the weakness (part C) against the organizational asset (part A)

Information learned while developing the loss scenarios helps to complete Part D of the risk scenario development, as depicted in Figure 4. By determining the various adverse impacts that might occur – whether by intentional attacks, natural events, or inadvertent errors – the enterprise will be able to support effective assessment, response, communications, and monitoring of information security risks. Notably, the goal is not to determine the probability that such a risk could occur since that exercise is part of risk analysis. Rather, the analysis of business impact is to determine what the various effects might be in order to enable risk managers to decide how critical and sensitive a particular business system is. Similar considerations apply to cyber-physical systems and operational technologies. The risk management process relies on this foundation of asset categorization, enabling a tailored and cost-effective approach to balancing risk and reward. Business impact drives categorization (sometimes called asset classification), which drives risk identification, which will later inform risk response, risk monitoring, and communication.

My recommendation is to follow the EBIOS RM methodology, from French ANSSI. The fourth workshop will aim at defining the "offensive scenarios" that are relevant for the environment for which you are running the risk management methodology. Those offensive scenarios should be considered as TTP (even if they are not directly referenced in MITRE ATT&CK Enterprise matrix), to be monitored by the SOC.

Focus your SOC detection engineering taskforce on priorities:

- Set as priority the detection of confirmed attack scenarios (and the corresponding TTP), as per risk management analysis.

Common detection use cases

On top of community SIEM rules, I wanted to highlight the following ones, that I consider as efficient based on experience. Threshold may need to be adapted to every context, obviously.

Detection logics

XDR-like detection logics:

- Correlation between EDR alert and CASB alert for the same endpoint, per timeframe.
- Correlation between EDR alert and CASB alert for the same user, per timeframe.
- Correlation between EDR alert and NDR alert for the same endpoint, per timeframe.
- Correlation between EDR alert and NDR alert for the same user per timeframe.
- Correlation between EDR alert and proxy SaaS alert for the same endpoint, per timeframe.
- Correlation between EDR alert and proxy SaaS alert for the same user, per timeframe.
- Correlation between EDR alert and identity management (AD, AAD, etc.) alert for the same user, per timeframe.

Threat intel-based detections:

- IOC match (C&C intel) on proxy SaaS logs, firewall logs, EDR logs (telemetry).
- IOC match on outgoing blocked traffic (FW, proxy, CASB), potentially indicating C&C traffic.

Unblocked infection vector:

- X EDR/antimalware detections for the same user, per timeframe (trying to detect an unblocked infection vector).
    - for instance, X > 2.
- X EDR/antimalware detections for the same workstation, per timeframe (trying to detect an unblocked infection vector).
    - for instance, X > 2.
- X EDR/antimalware detections for the same server, per timeframe (trying to detect an unblocked infection vector).
    - for instance, X > 9 (NB: might need to be higher for file sharing servers).

Persistance or protection bypass capabilities of threat:

- EDR/antimalware cleaning error.

- EDR/antimalware detection during scheduled scan (meaning the threat has bypassed realtime protection).
- A phishing URL has been clicked on before it was detected (Eg.: MS 365 Defender and ProofPoint UrlDefense offer this detection capability).

Successful vulnerability exploitation detection:

- Correlation of firewall logs (outgoing traffic) and a list of IP addresses that are sources of detected attacks by WAF and NIDS;
    - NB: this is most likely a hint that a vulnerability has successfully been exploited and there is a callback to an attacker's machine.

Impossible scenarios:

- Same user authenticating within X min of timeframe, on two different endpoints (workstations/mobiles, not being located in the same place);
    - for instance, X > 2min.
- Same user (except admins, to begin with) authenticating on more than X endpoints (workstations/mobiles), per timeframe (eg.: 10 min);
    - for instance, X > 2.

Successful bruteforce [MITRE T1110]:

- Same user having X wrong passwords followed by successful authentication;
    - for instance, X > 100
    - See this Splunk Webinar, page 38.

Lateral movement [MITRE T1021.001]:

- Multiple RDP servers to which an user connects over RDP for the first time;
    - See this Splunk Webinar, page 33.

C&C activity [MITRE T1071.004]:

- C2 beaconing over DNS:
    - See this Splunk article, and this one;
    - See this blog article;
    - See this presentation, hypothesis #2.

Newly accessed domains:

- Typically landing page for infection, or C2C;
    - See this Splunk article
    - NB: you may want to query all of the query results onto your TIP, leveraging automation capabilities (SOA). Thus, you will prioritize the handling of those network traffic logs.

Potential information leak:

- Detect abnormal traffic peaks, within the outgoing traffic logs (FW, proxies);
    - See this Splunk presentation

Obfuscated script [T1027, T1059]:

- Typically obfuscated PowerShell with base64;
    - See the Splunk's Git
    - If you wanna go further, see this Splunk article

Augmenting detection with automation

See threat intel page

Everything-as-code (DevSecOps)

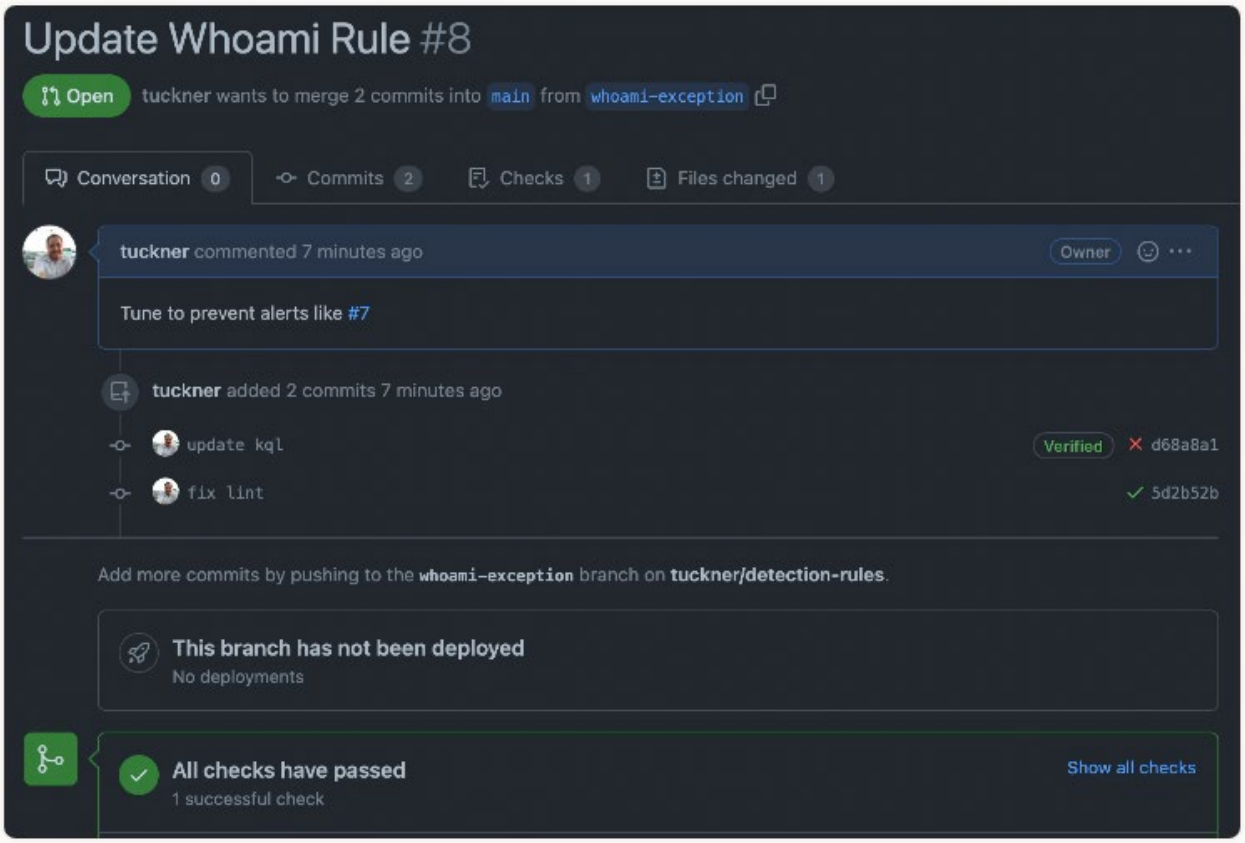
The idea here is to follow the 'as-code' approach, wherever possible, with a central repository as a versioning system and source of truth. This, in order to achieve automation, quality controls, resilience (restore previous version in case something breaks), R&D with PDCA, etc. For instance, based on experience, this is applicable to SIEM rules, SOA playbooks, SOP, etc.

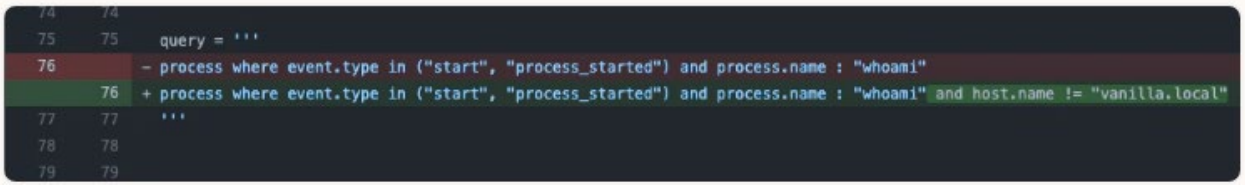
Required tools

- My recommendation: GitLab (or equivalent)

Detection-as-code

- Implement CI/CD/CD between the SIEM rules and an internal Git repository;
    - My recommendation: use YAML to declare and store your SIEM rules.
    - See [example] here with Elastic and Git





- Implement CI/CD/CD between the SIEM apps and an internal Git repository.
- Implement CI/CD/CD between the SIEM objects templates (if any) and an internal Git repository.
- Implement CI/CD between the audit policies (e.g.: Sysmon XML files, Linux AuditD conf, ...) and an internal Git repository.

Response-as-code

- Implement CI/CD/CD between the SOA and an internal Git repository, for the playbooks;
    - Store the playbooks on the Git repo.
    - Version them thanks to Git (test / preprod / prod / disabled).
    - Deploy them from the Git, as a point of reference, onto the SOA.

SOP-as-code

- Implement CI/CD/CD between the SOP (Standard Operating Procedures) hosted on a Wiki (or equivalent) and and internal Git repository;
    - My recommendation, to host the documentation and SOP: GitLab Docs.

To go further

Must read

- Synaktiv, Traces of Windows Remote Command Execution
- Awesome Detection Engineering.
- Naksyn, Operating into EDRs blindspot
- MAGMA, use case management framework.
- Palantir, ADS Framework, Alerting and Detection Strategies framework.
- Splunk, Detection Deep dive.
- NIST, IR 8286D: BIA and Risk Prioritization

End