One step Closer to Detection Engineering
The Linchpin Framework

Atanas Viyachki (he/him)
CISSP | GCIH | GPYC | Senior Threat Hunter@Wealthsimple

Agenda

1. **Welcome** (1 minutes)
2. **Intro slides** (5 minutes)
3. **Framework Concepts** (15 minutes)
4. **Applying ODEF (**15 minutes)
5. **Summary & Conclusions** (surplus time!)
6. **Thank you!**


?
Quiz
Time

Do you write custom detections in your organization?

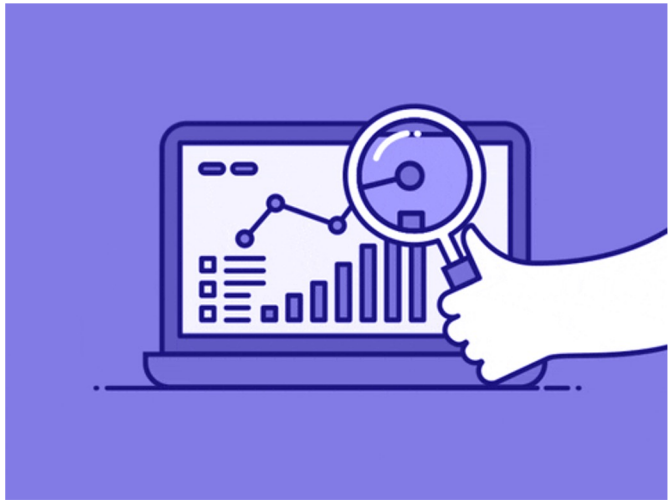
Do you have detection engineers?

Are the detection engineers solely dedicated on writing detections?

What is **D**etection **E**ngineering?

Detection Engineering is a capability that **researches** and models threats in order to deliver modern and effective threat detections.

Goal is to provide ***automated*** analytical capabilities {***hunts***} that can capture and detect the ***behaviours*** and TTPs of adversaries.



Gif source: https://giphy.com/gifs/analytics-pixelmonkeys-tony-babel-l46Cy1rHbQ92uuLXa

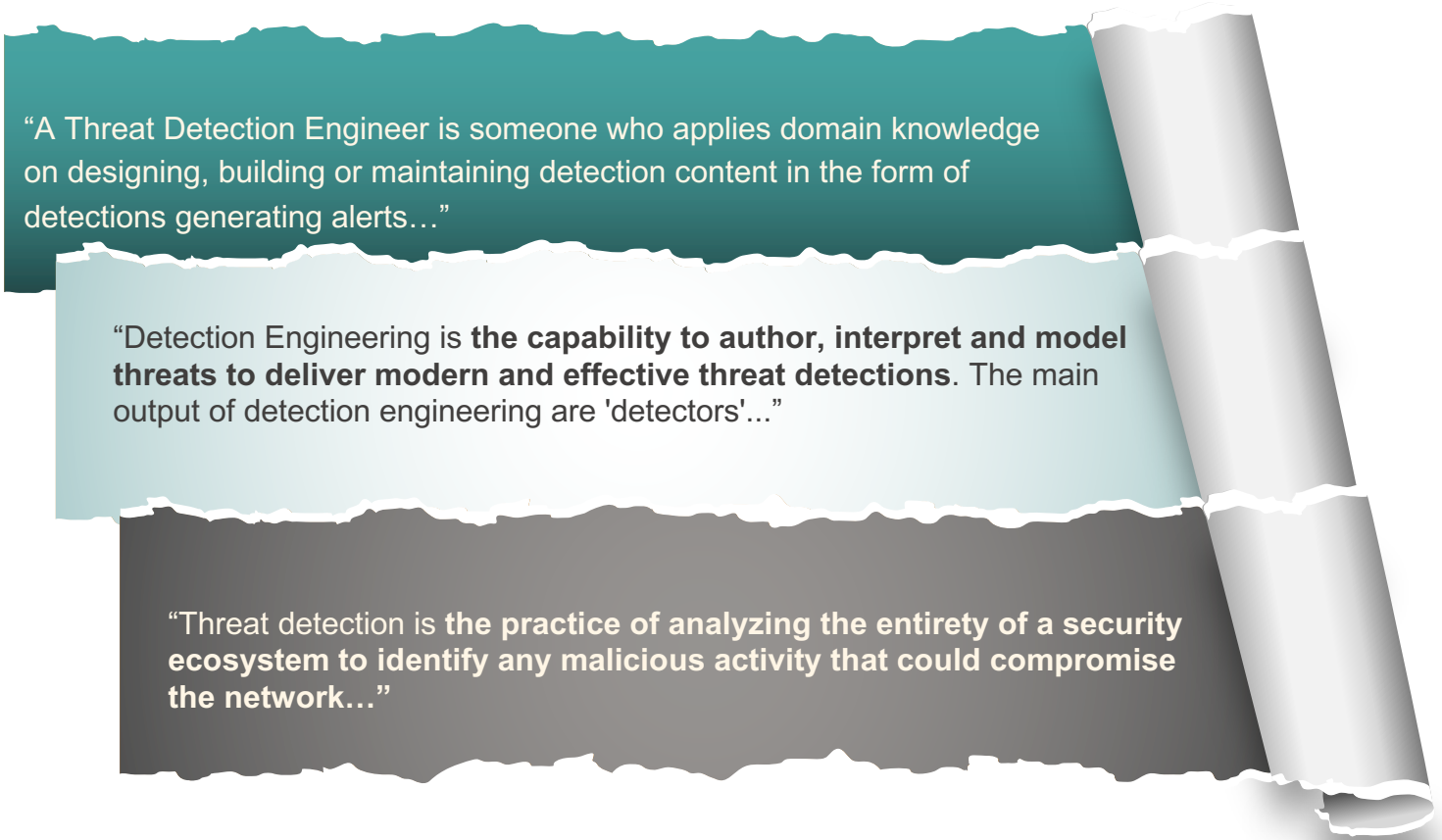
Why developing detections in house when we have vendors?

Security vendors are good and we need them
however they don't…

…***know our environment*** like we do
…necessarily ***have tailored intel*** for each client
…***disclose*** their security ***rules*** and detection ***logic***

As a defenders, we can do better and provide more ***assurance.***

How the world sees Detection Engineers?



My vision for DE

With a good **cookbook!!!**

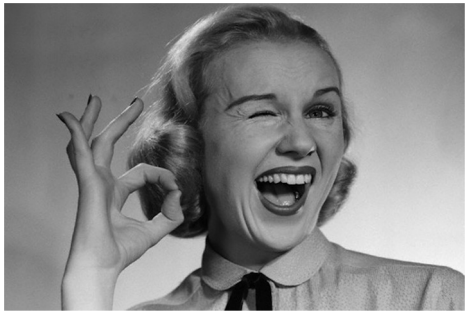


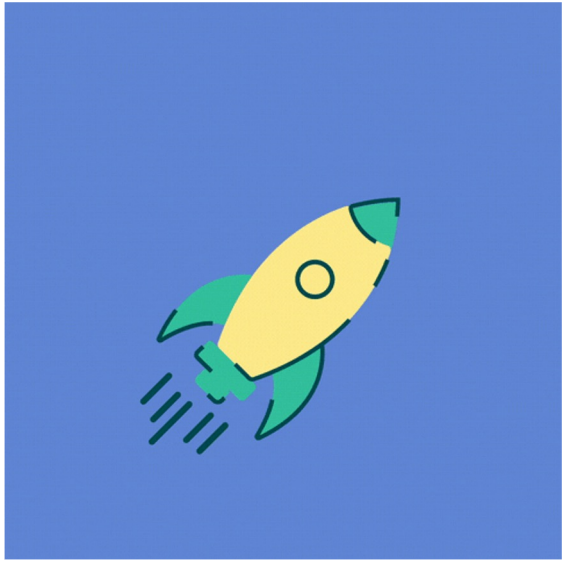
Image sources: http://www.danthepixarfan.com/2019/06/ratatouille-anyone-can-cook-journal.html
https://giphy.com/search/ok

Requirements

1. ***Framework*** that provides guidelines and enables ***any team*** to efficiently ***build high quality detections*** and deliver best in class ***documentation.***

1. Technical implementation that will ***centralize***, ***standardize*** and ***preserve*** the ***knowledge*** that our ***security community*** have and enable knowledge ***sharing***.

1. Naturally grow to a ***Detection as a Code*** capability



Why use cookbooks at all?

THIS

NOT THAT

Why use ~~cookbooks~~ frameworks at all?

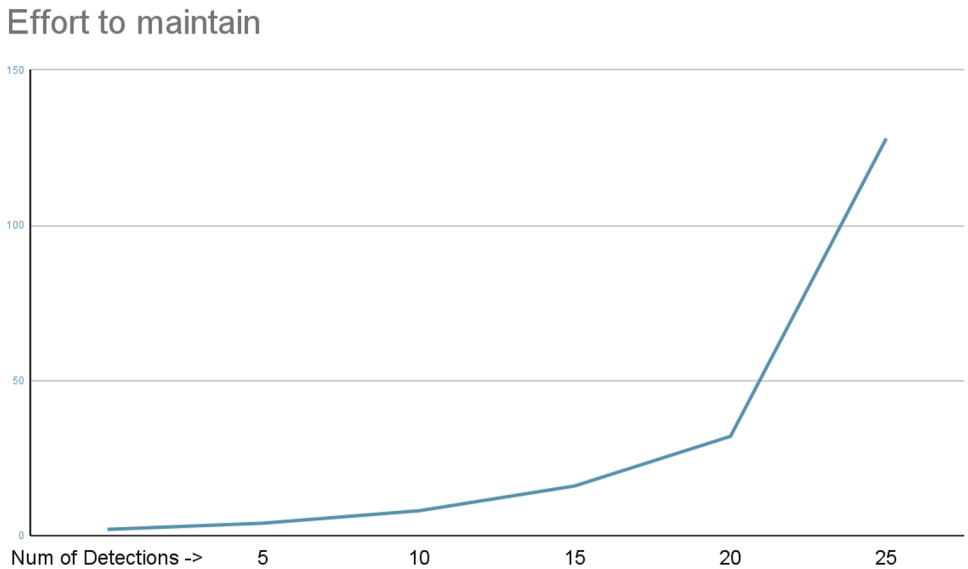


Image sources: https://www.google.com/search?q=scalability&client=firefox-b-d&sxsrf=ALiCzsbkre2POdkZbuY3YMwGkr5UpfPiKw:1661094551021&source=lnms&tbm=isch&sa=X&ved=2ahUKEwinnpu1m9j5AhWMjlkEHbI2D4MQ_AUoAXoECAIQAw&biw=1472&bih=744&dpr=2.61#imgrc=4HYwgp5S2dmWfM&imgdii=zzz9xg2ALzcRTM

The Open Detection Engineering Framework
Part 1: Concepts

Open **D**etection **E**ngineering **F**ramework - **ODEF**

Benefits for End-Users 

- Provides ***guidelines*** on how to be ***systematic, repeatable*** and ***thorough*** while you engineer detections
- Converts insights to retainable and actionable ***knowledge*** and promotes knowledge ***sharing***
- Introduces detection ***lifecycle*** and ***maturity***

Fundamentals of the ODEF

Framework Core

- Detection lifecycle phases:
  - Sunrise
  - Midday
  - Sunset

- Phase components:
  - Functions
  - Goals
  - Guidelines

Detection Engineering Maturity Model (DEMM

- Maturity levels:
  - Partial
  - Adequate
  - Enabled

ODEF Phases

Phase → **Sunrise**

Functions →

- **Research** until you know
- **Prepare** with logging and visibility
- **Build & Enrich** high fidelity detection with a baseline
- **Validate** the detection
- **Automate** with your SOAR/SIEM
- **Share** the detection in the security organization

ODEF Phases

Phase →

Midday

Functions →

- **Monitor** the detection
- **Improve** the detection logic in case of influx of FP
- Perform systematic **reviews** to ensure detection relevancy

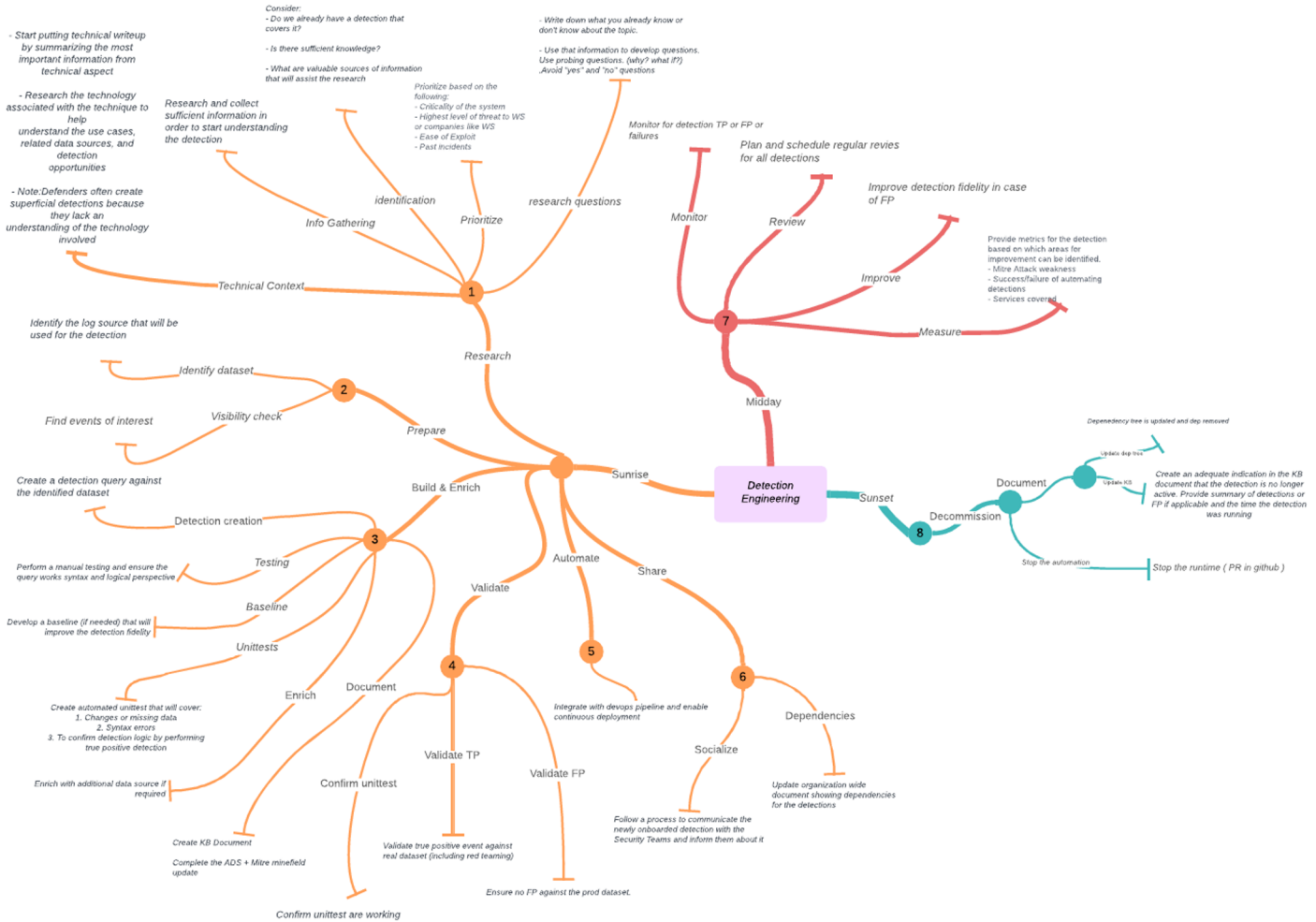
ODEF Phases

Phase →

Sunset

Functions →

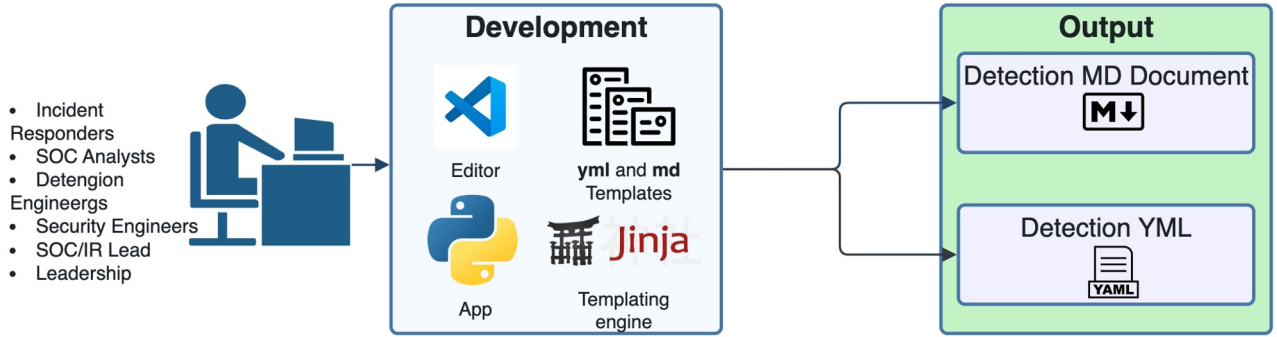
- **Decommission** and leave it in a state that it can be resumed anytime
- **Preserve knowledge**


Detection Engineering
Sunrise
Midday
Sunset
Research
Prepare
Build & Enrich
Automate
Share
Validate
1
2
3
4
5
6
7
8
Technical Context
- Start putting technical writeup by summarizing the most important information from technical aspect
- Research the technology associated with the technique to help understand the use cases, related data sources, and detection opportunities
- Note:Defenders often create superficial detections because they lack an understanding of the technology involved
Info Gathering
Research and collect sufficient information in order to start understanding the detection
identification
Consider:
- Do we already have a detection that covers it?
- Is there sufficient knowledge?
- What are valuable sources of information that will assist the research
Prioritize
Prioritize based on the following:
- Criticality of the system
- Highest level of threat to WS or companies like WS
- Ease of Exploit
- Past incidents
research questions
- Write down what you already know or don't know about the topic.
- Use that information to develop questions. Use probing questions. (why? what if?) Avoid "yes" and "no" questions
Identify the log source that will be used for the detection
Identify dataset
Visibility check
Find events of interest
Create a detection query against the identified dataset
Detection creation
Testing
Perform a manual testing and ensure the query works syntax and logical perspective
Baseline
Develop a baseline (if needed) that will improve the detection fidelity
Unittests
Create automated unittest that will cover:
1. Changes or missing data
2. Syntax errors
3. To confirm detection logic by performing true positive detection
Enrich
Enrich with additional data source if required
Document
Create KB Document
Complete the ADS + Mitre minefield update
Confirm unittest
Confirm unittest are working
Validate TP
Validate true positive event against real dataset (including red teaming)
Validate FP
Ensure no FP against the prod dataset.
Integrate with devops pipeline and enable continuous deployment
Socialize
Follow a process to communicate the newly onboarded detection with the Security Teams and inform them about it
Dependencies
Update organization wide document showing dependencies for the detections
Monitor
Monitor for detection TP or FP or failures
Review
Plan and schedule regular revies for all detections
Improve
Improve detection fidelity in case of FP
Measure
Provide metrics for the detection based on which areas for improvement can be identified.
- Mitre Attack weakness
- Success/failure of automating detections
- Services covered
Decommission
Document
Dependency tree is updated and dep removed
Update dep tree
Update KB
Create an adequate indication in the KB document that the detection is no longer active. Provide summary of detections or FP if applicable and the time the detection was running
Stop the automation
Stop the runtime (PR in github)

Part 2: Applying ODEF
A detection engineering story

Utilize templates

Goal:



Benefits of using templates:

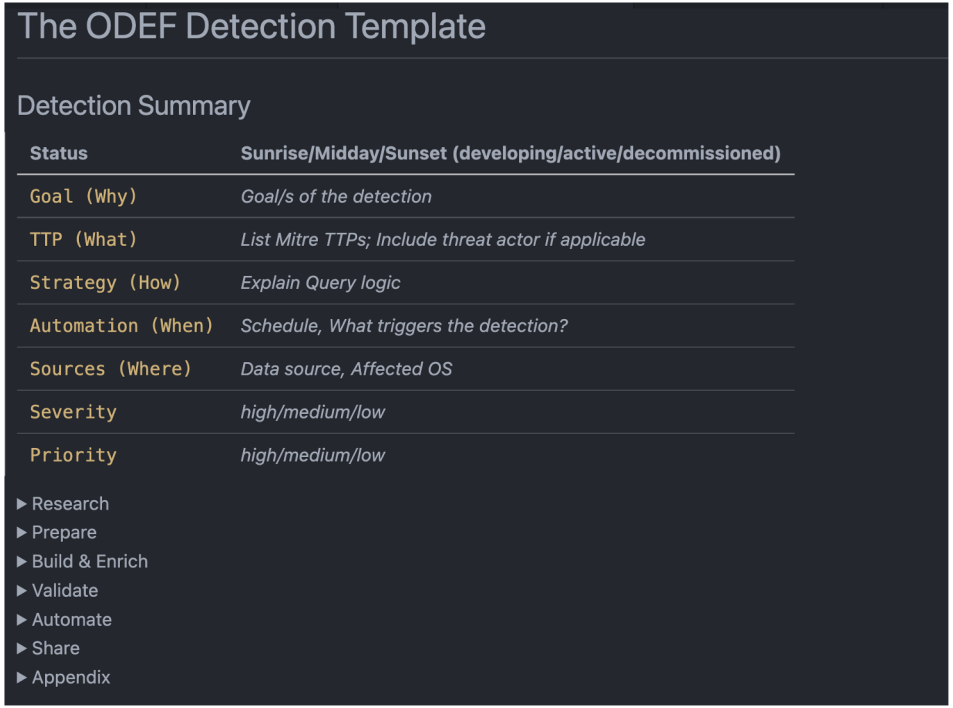
- Single language for all security teams
- Easy to read documentation
- Ensures quality and minimizes mistakes
- Standardized output

The ODEF Detection Template

Goal: Structure and retain the knowledge collected during research and make it shareable across the organization.

Benefits:

- Improves documentation quality
- Enforces a detection creation process **aligned with sunrise phase**
- Provides consistency and uniformity



The ODEF Detection Template

Detection Summary

Status	Sunrise/Midday/Sunset (developing/active/decommissioned)
Goal (Why)	*Goal/s of the detection*
TTP (What)	*List Mitre TTPs; Include threat actor if applicable*
Strategy (How)	*Explain Query logic*
Automation (When)	*Schedule, What triggers the detection?*
Sources (Where)	*Data source, Affected OS*
Severity	*high/medium/low*
Priority	*high/medium/low*

▶ Research
▶ Prepare
▶ Build & Enrich
▶ Validate
▶ Automate
▶ Share
▶ Appendix

The ODEF Detection 

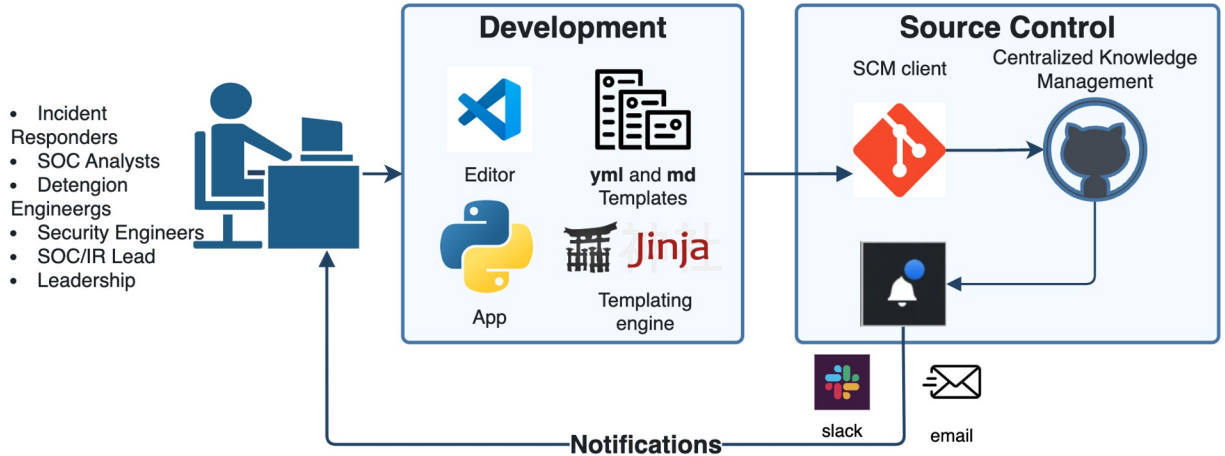
Goal: Standardize and **abstract** the detections syntax independently of the platform

Benefits

- Minimizes errors by allowing for better readability and troubleshooting
- Standard structure can be used to **convert** from yml to other standards (terraform, json etc)
- yml is tested with python-cerberus framework

```yaml
status: sunrise/midday/sunset
created_date: "2023-03-23"
last_review_date:
name: Detection_Name
query: Detection query
baseline:
- string which entails your baseline
- Note, if multiple baselines
- Add them one by one using hyphen
visualization: Describe the fields that are being visualized as output of the query
schedule: For example, ever 1h, or 1 week etc.
event_limit: 0
data_source: The data source used.
data_location: The location of the data - cloud, on-prem, system name, bucket name etc.
tactic: tactic covered with the detection
mitre_id: Mitre id, for example->T1563.001
mitre_url: https://attack.mitre.org/techniques/T1098/001/
detection_severity: Severity is a measurement of impact in case of true positive.
incident:
  priority: numeric, 0-unknown, 0.5 - informational, 1-low,2-medium,3-high,4-critical
  type: type of the incident/malware/hacking etc.
  name: incident name - normally sourced from the detection name
  description: 'The human readable description of the incident'
  sla: if applicable
```

Enhance with Source Control



Benefits of using SCM:

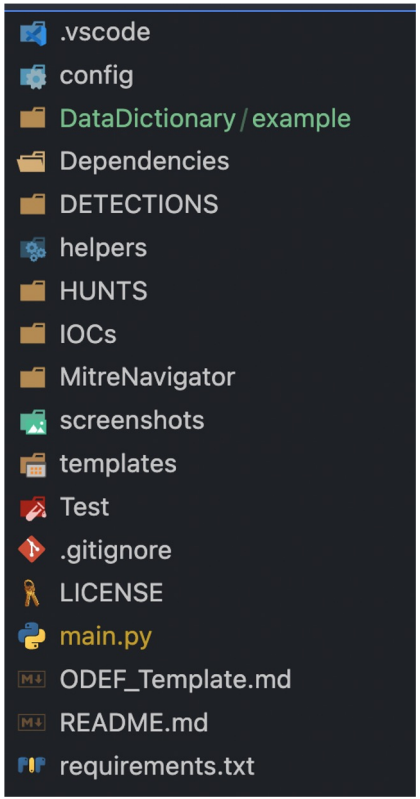
- Change control
- Enables **knowledge sharing**
- Standardized content structure
- Awareness through notification
- More efficient collaboration

= GitHub notifications

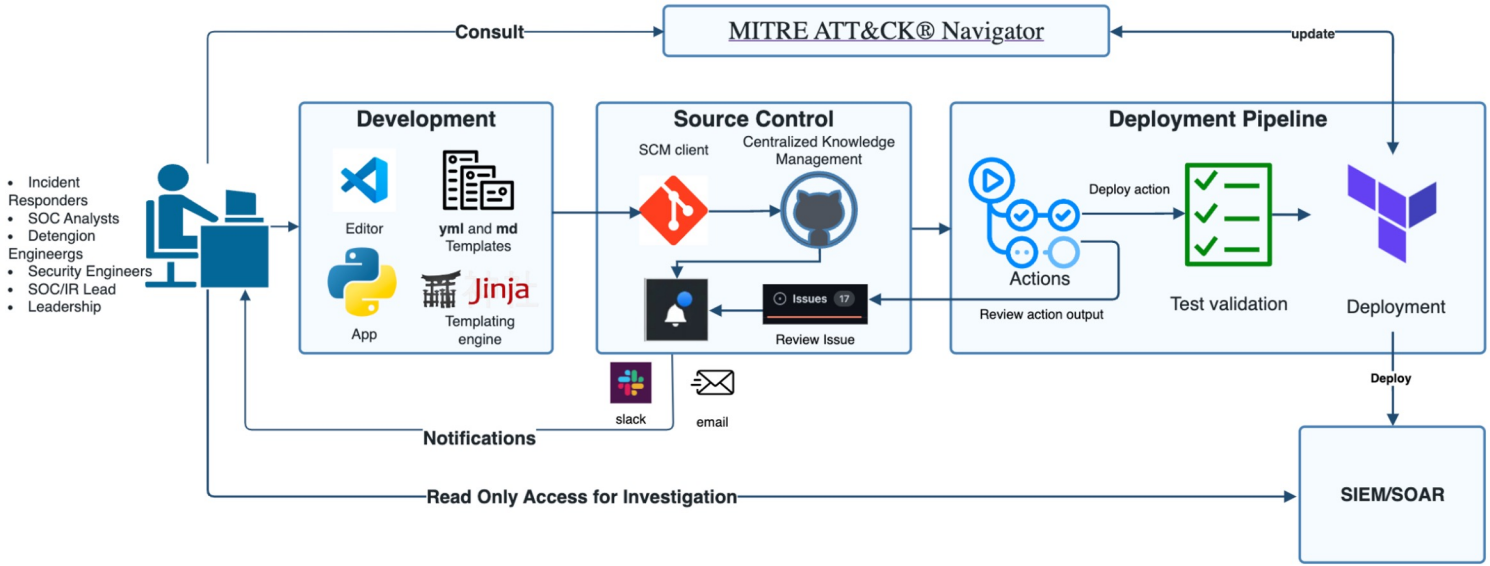
SCM = Source Control Management

Content Organization

- Detections & Hunts & IOCs
  - Folders providing structure for documenting and storing your security content
- Templates
  - Contains templates used by the python app for detection yml
- DataDictionary
  - Any data set used can be documented and included in the data dictionary
- Dependencies
  - Folder that provides birdseye view of the data sources and fields used in detections and hunts
- MitreNavigator input
  - Contains the json file used to visualize your defensive coverage
- Tests
  - Unittests to check the validity of the detection yml files



Grow to Detection as Code



Benefits DaC:

- Unit testing
- Modification Control
- Fail-safe deployment
- Reporting
- Consistency

SIEM = Security Information and Event Management
SOAR = Security Orchestration, Automation and Response
DaC = Detection as Code

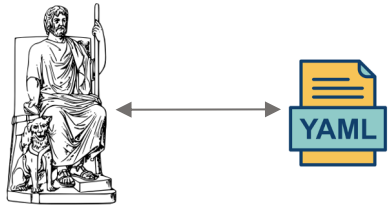
Quality Assurance

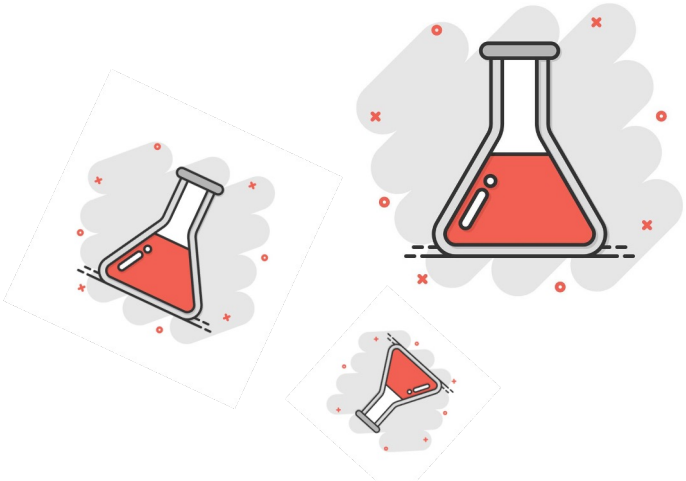
Unittests

- File content validation
- File exist validation

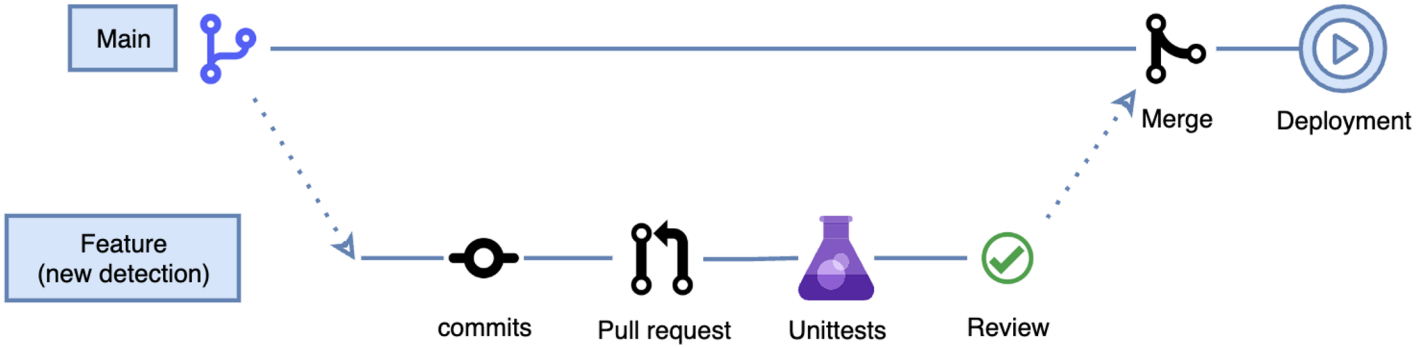
Cerberus - The watchdog of Hades

- Uses a schema for yml content validation





PR Review Process



Enforcing Detection Reviews via Automation

Implementation with:

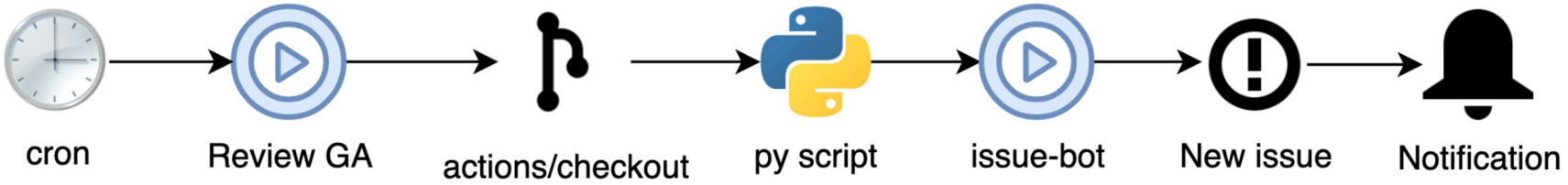
- Github Action
- Python
- Github Issues
- Github Notifications

In the config:

In the detection yml:

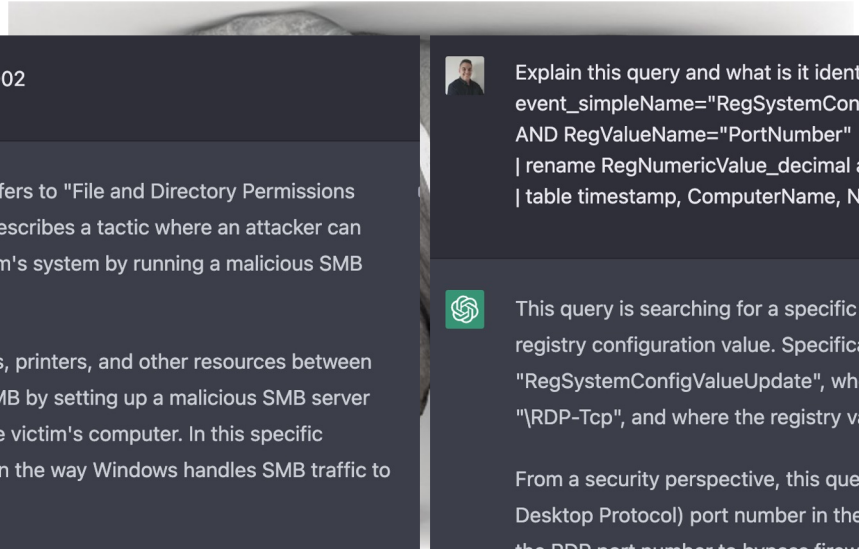
```
status: sunrise/midday/sunset
created_date: "2023-03-23"
last_review_date:
name: Detection_Name
```

Process flow



The future

Using ChatGPT API for context enrichment



Explain MITRE attack technique with id T1222.002

MITRE ATT&CK technique with id T1222.002 refers to "File and Directory Permissions Modification by Malicious SMB Server" and it describes a tactic where an attacker can modify file and directory permissions on a victim's system by running a malicious SMB (Server Message Block) server.

SMB is a network protocol used for sharing files, printers, and other resources between computers. Attackers can take advantage of SMB by setting up a malicious SMB server that can send specially crafted messages to the victim's computer. In this specific technique, the attacker exploits a vulnerability in the way Windows handles SMB traffic to modify the permissions on files and directories.

The attacker sends a crafted message to the victim's computer, which results in the malicious SMB server modifying the file and directory permissions on the victim's system. The attacker can then use the modified permissions to access sensitive files, execute malicious code, or escalate privileges on the system.

Explain this query and what is it identifying from security perspective:

```
event_simpleName="RegSystemConfigValueUpdate" AND RegObjectName="*\RDP-Tcp"
AND RegValueName="PortNumber"
| rename RegNumericValue_decimal as "NewRDPPort"
| table timestamp, ComputerName, NewRDPPort
```

This query is searching for a specific type of event that is related to the update of a registry configuration value. Specifically, it is looking for events with a simple name of "RegSystemConfigValueUpdate", where the registry object name contains the string "\RDP-Tcp", and where the registry value name is "PortNumber".

From a security perspective, this query is looking for changes to the RDP (Remote Desktop Protocol) port number in the Windows registry. Attackers may attempt to change the RDP port number to bypass firewall rules or to avoid detection by security tools that are only monitoring the default RDP port (TCP port 3389). By monitoring changes to this registry value, security teams can detect potential malicious activity and take appropriate actions to mitigate the threat.

https://chat.openai.com/chat

Summary & Conclusion

1. Start small and build big.
2. Engage the power of your security tribe.
3. Contextualize your detections and aim to grow to ‘Detection as Code’ capability.

Resources

- **ODEF public repository**: https://github.com/wealthsimple/odef
  - Framework: https://github.com/wealthsimple/odef/blob/main/Docs/ODEF-README.md
  - ODEF Detection template: https://github.com/wealthsimple/odef/blob/main/templates/documentation-template.md
- Tools:
  - Mitre ATT&CK Navigator
  - Visual Studio Code
  - Terraform
  - Jinja
  - Markdown Cheatsheet
  - Cerberus

Thank you!

LinkedIn → https://www.linkedin.com/in/viyachki/

Email → atanas.viyachki@wealthsimple.com

ODEF Blog → https://medium.com/@aviyachki/part-1-bro-do-you-even-detection-engineer-1584dca5ddc9