

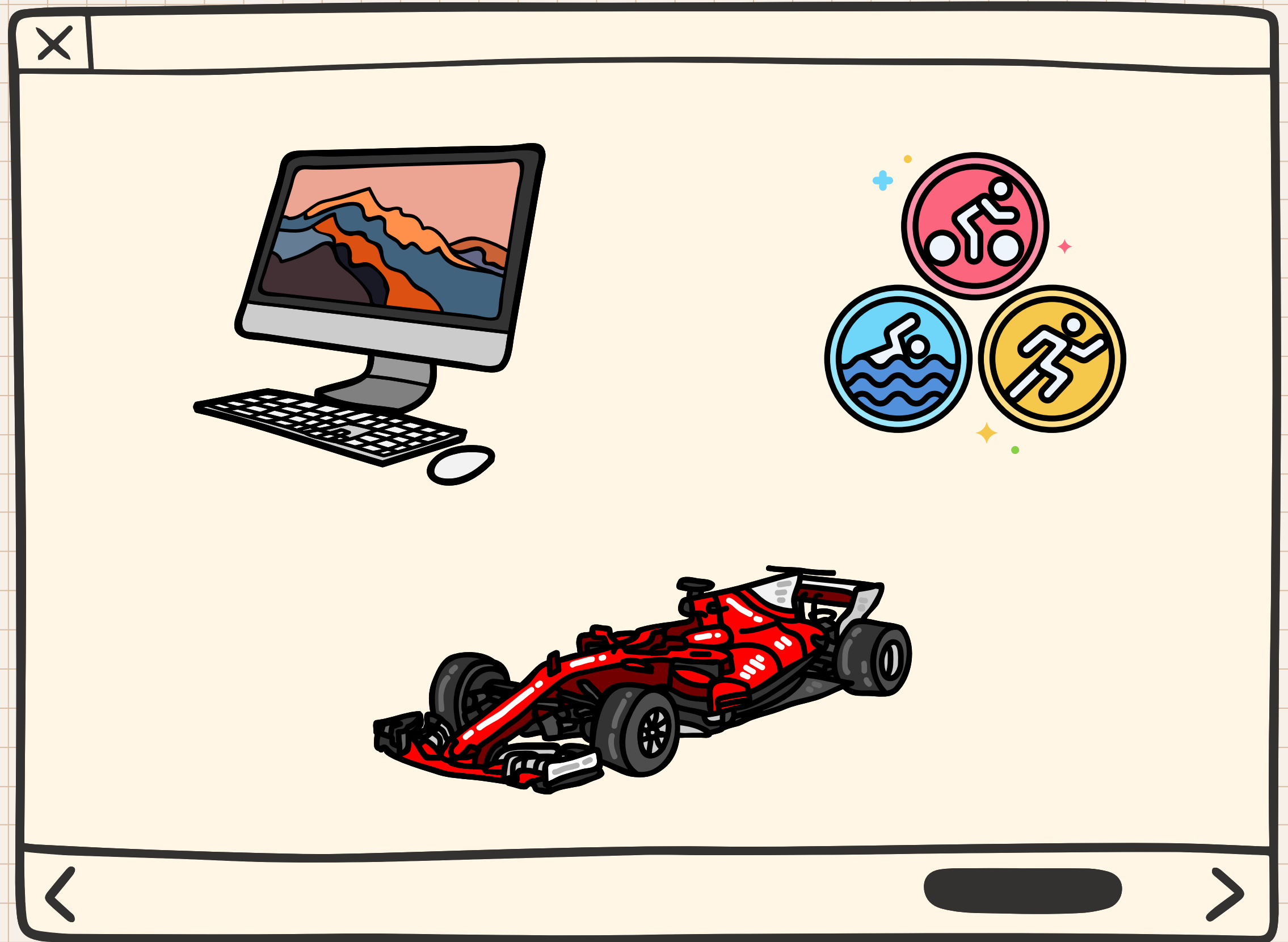


# The Detection Engineering Process

Applying the scientific method to  
Detection Engineering

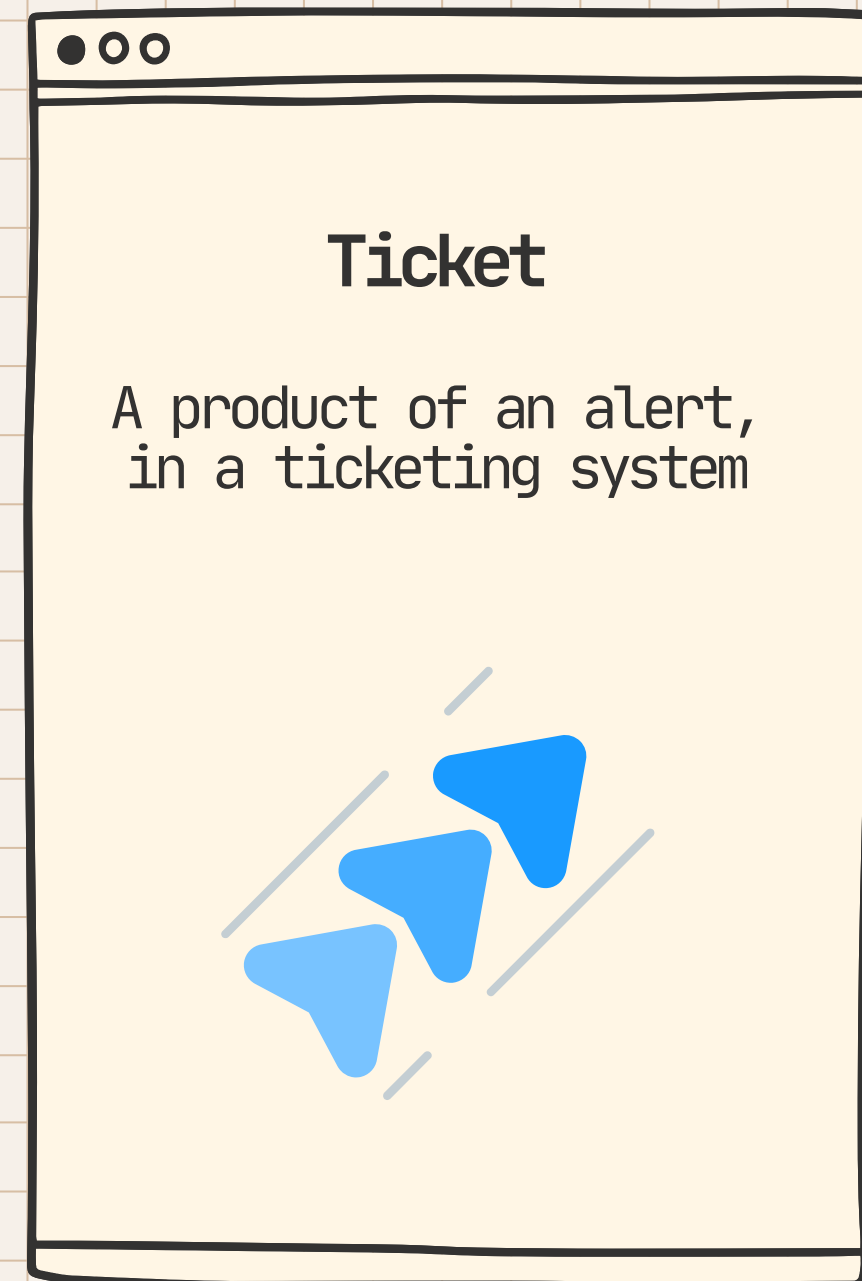
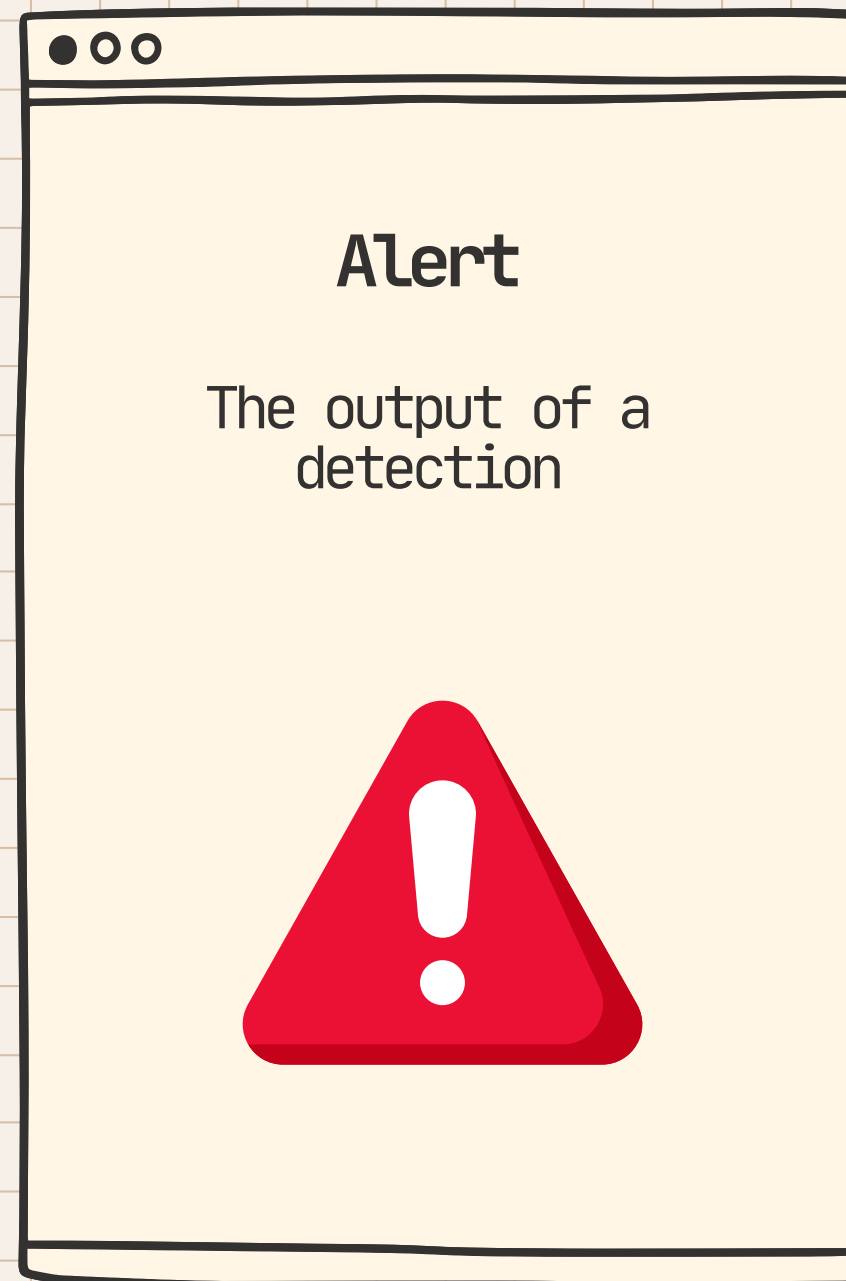
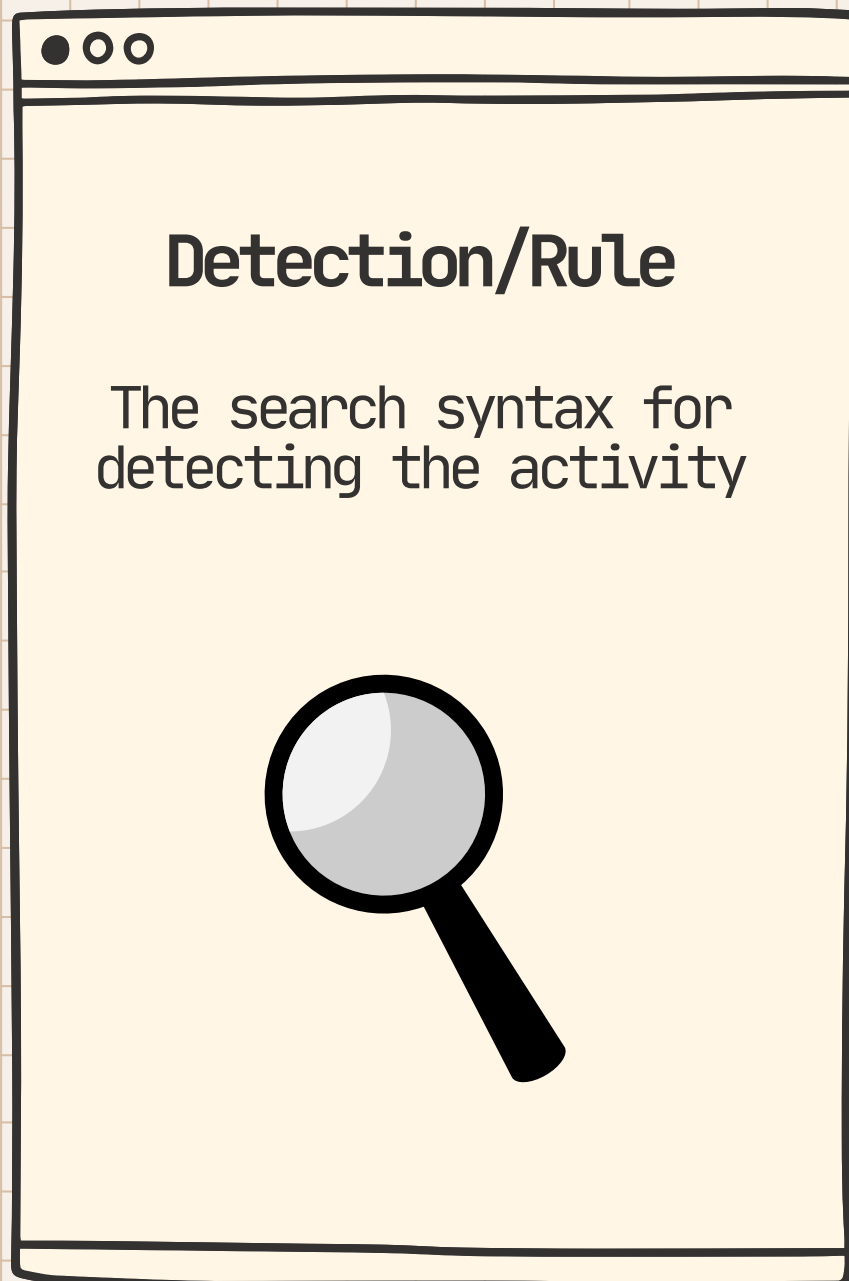
# \$ *whoami*

- The definition of Type A
- BHIS SOC
- Triathlete, engineer, and Formula 1 addict



# Detection Engineering

## *Key Terms*

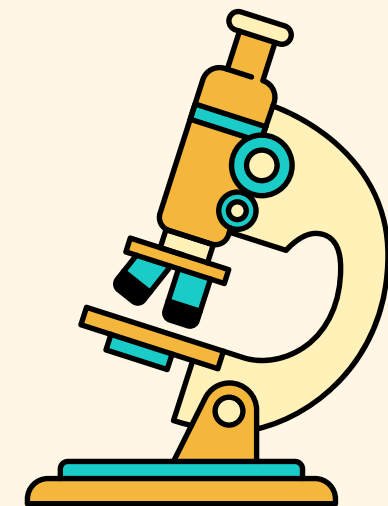


Detect -> Alert -> Ticket

# Scientific Method

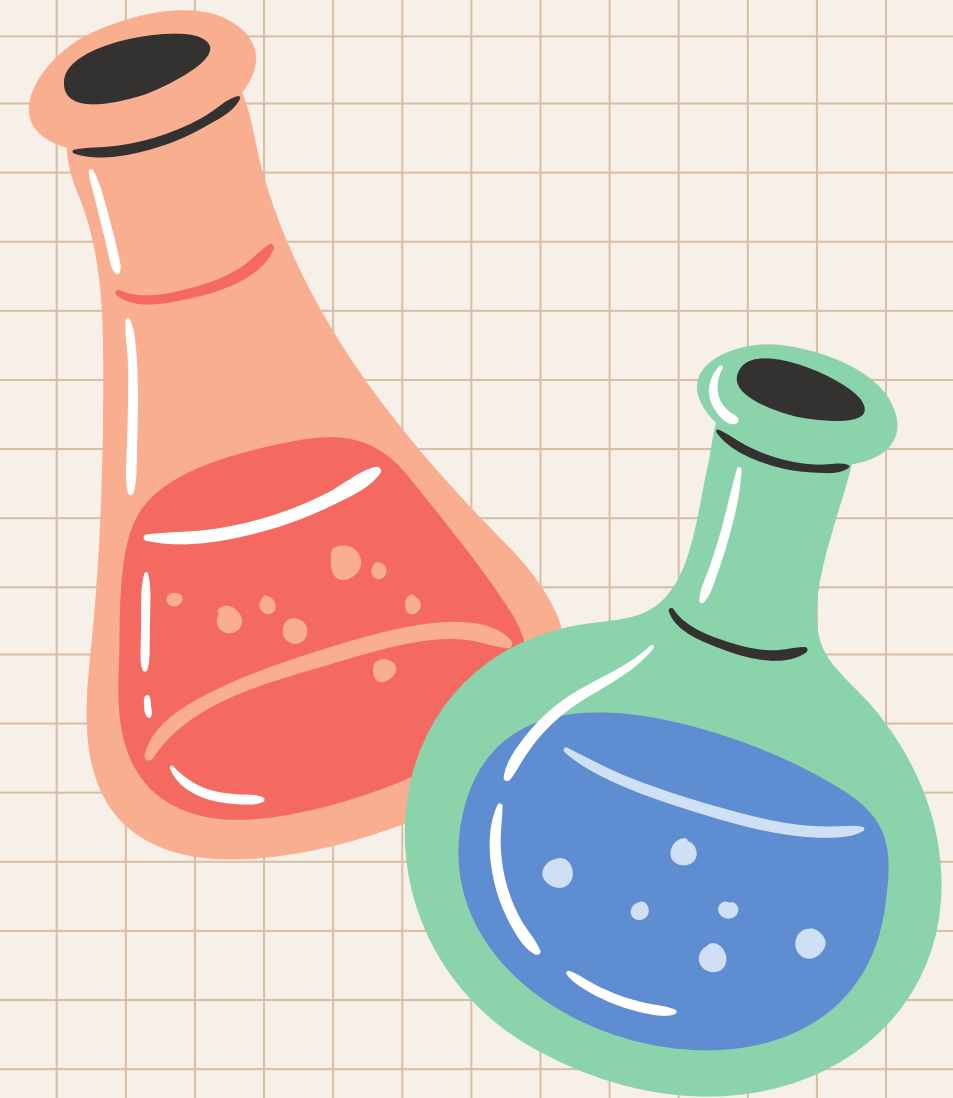
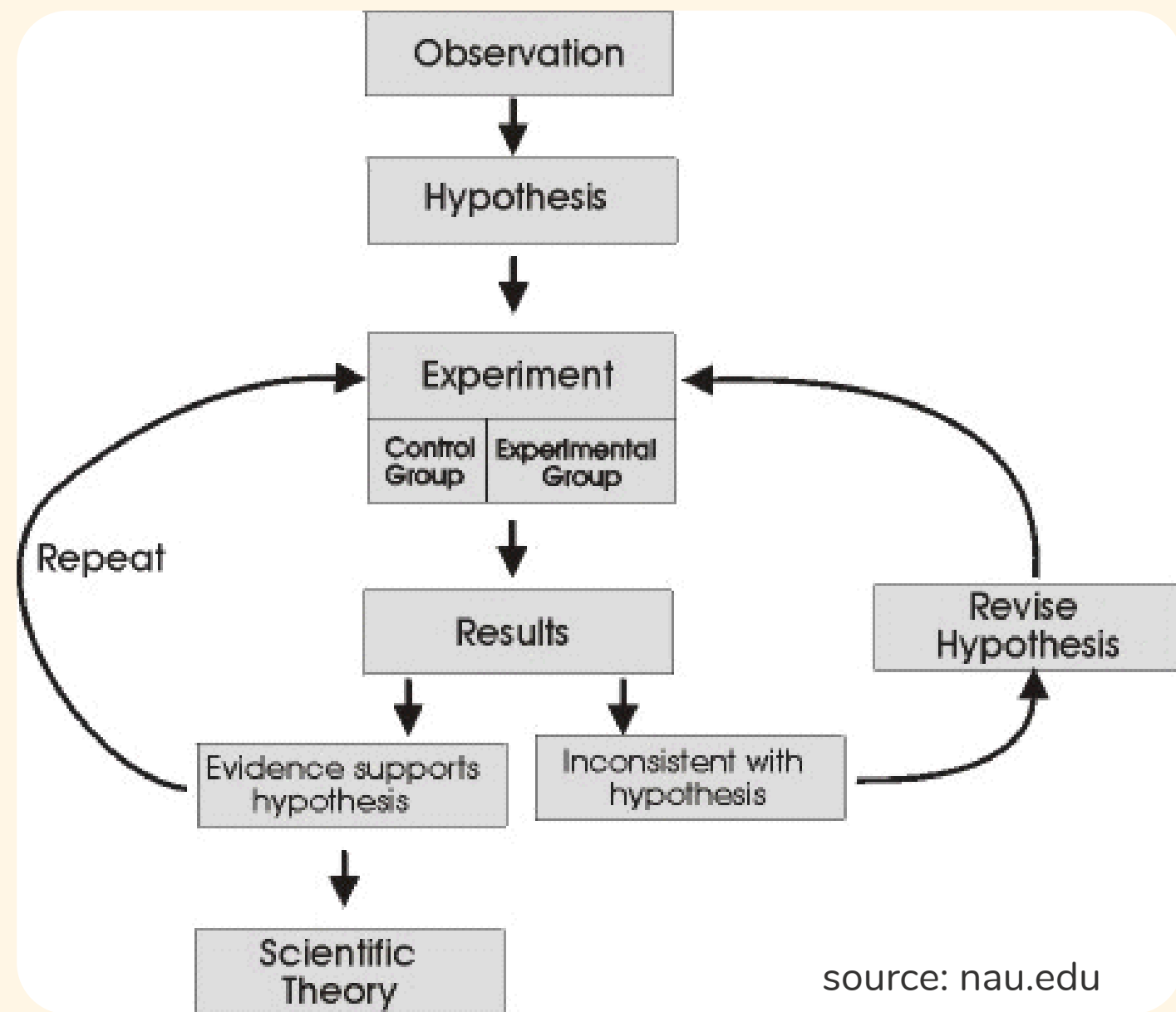
Science follows a method of rigor and observation to come to meaningful conclusions

1. Observation
2. Research
3. Hypothesis\*
4. Experimentation
5. Analysis
6. Report / Conclusions
7. Repeat



*\*Note that the process doesn't start here! Research has resulted in a hypothesis forming*

# Scientific Method

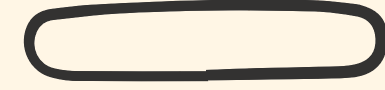
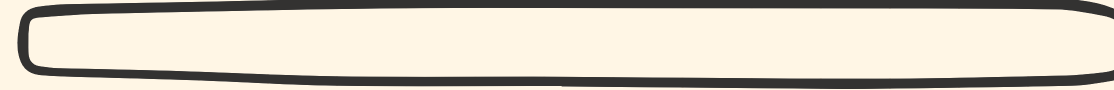
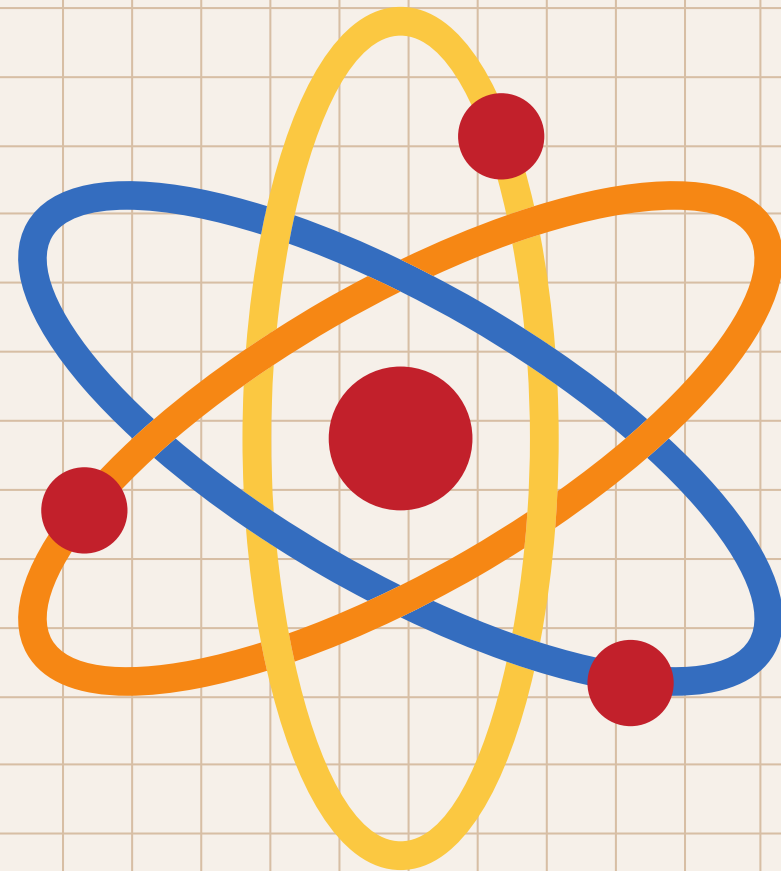




# Theories

“A scientific theory is an explanation of an aspect of the natural world and universe that **can be repeatedly tested and corroborated** in accordance with the scientific method”

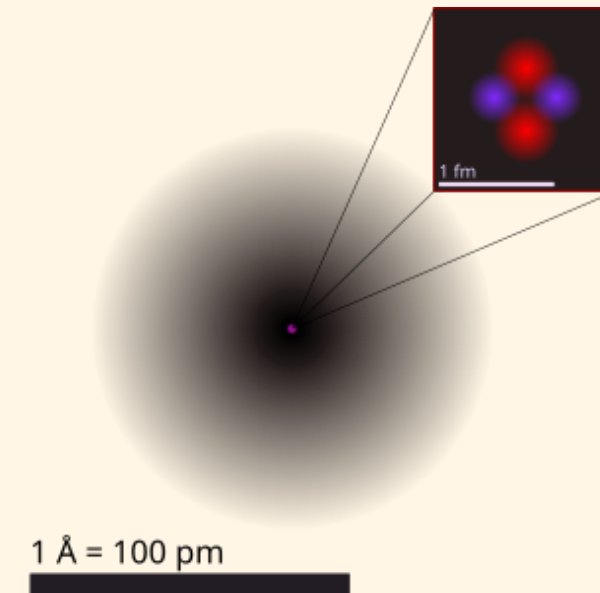
source: Wikipedia



## Atomic Theory

“The scientific theory that matter is composed of particles called atoms.”

Research, discoveries, and rigorous testing helped form this theory



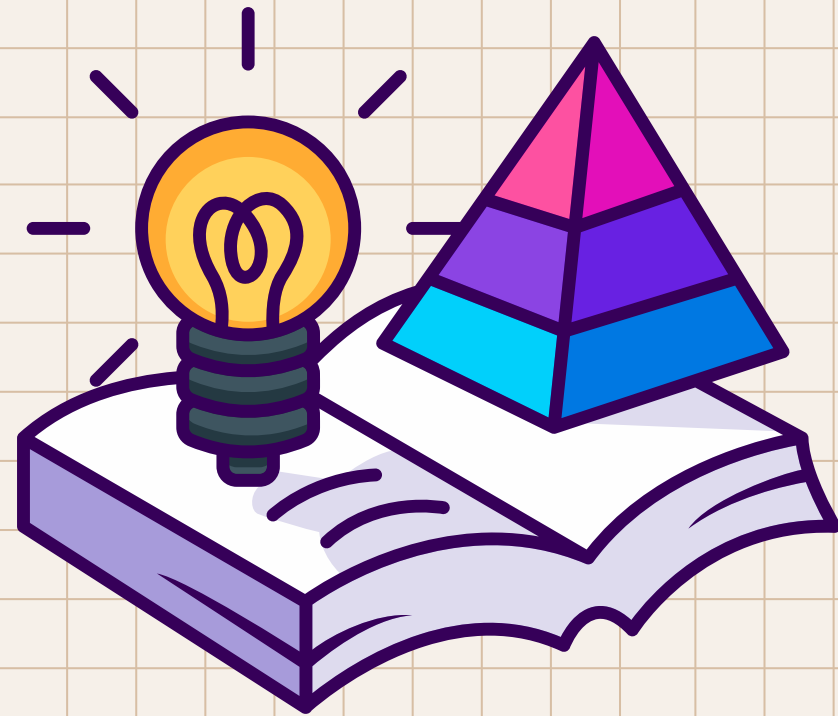
It did not remain the same from when it was first composed

source: Wikipedia



...

What is the equivalent to  
a theory in Detection  
Engineering?



< >  Q ×

## A High-Quality Detection

- Well-researched and documented
- High fidelity, low volume
- Verifiable and reproducible
- Continuous improvement

...

"What does this have to do  
with detection engineering?"  
- You, probably




...

"What is this? School? Can  
we talk about the actual  
topic?"  
- Also you, probably



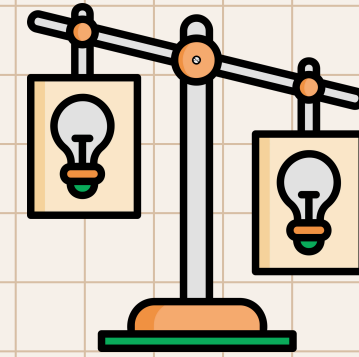
# Detection Engineering Process

Follows a similar flow if viewed correctly

- 
0. Detection Story
  1. Research\*
  2. Query
  3. Backtest
  4. Canary
  5. Documentation
  6. Onboarding
  7. Continuous Improvement

*\*The hypothesis could form here, if not operating off a detection story*

# Comparison



## Scientific Method

Observation  
Research  
Hypothesis  
Experimentation  
Analysis  
Conclusion  
Repeat

## Detection Engineering

Story  
Research  
Query  
Backtest  
Canary  
Documentation  
Onboarding  
Improvement

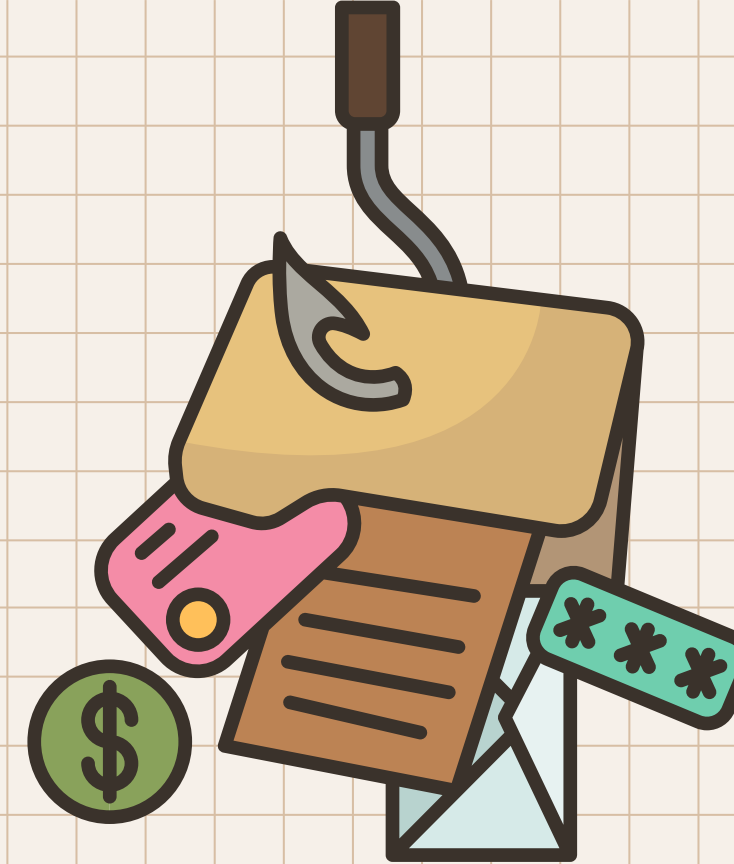
# Why?

What are the benefits of following a more rigorous process?

To name a few:

- Better defined scope
- Easier understanding of the detection
- No missed or forgotten steps/components
- Higher quality detections overall





# Let's Apply It



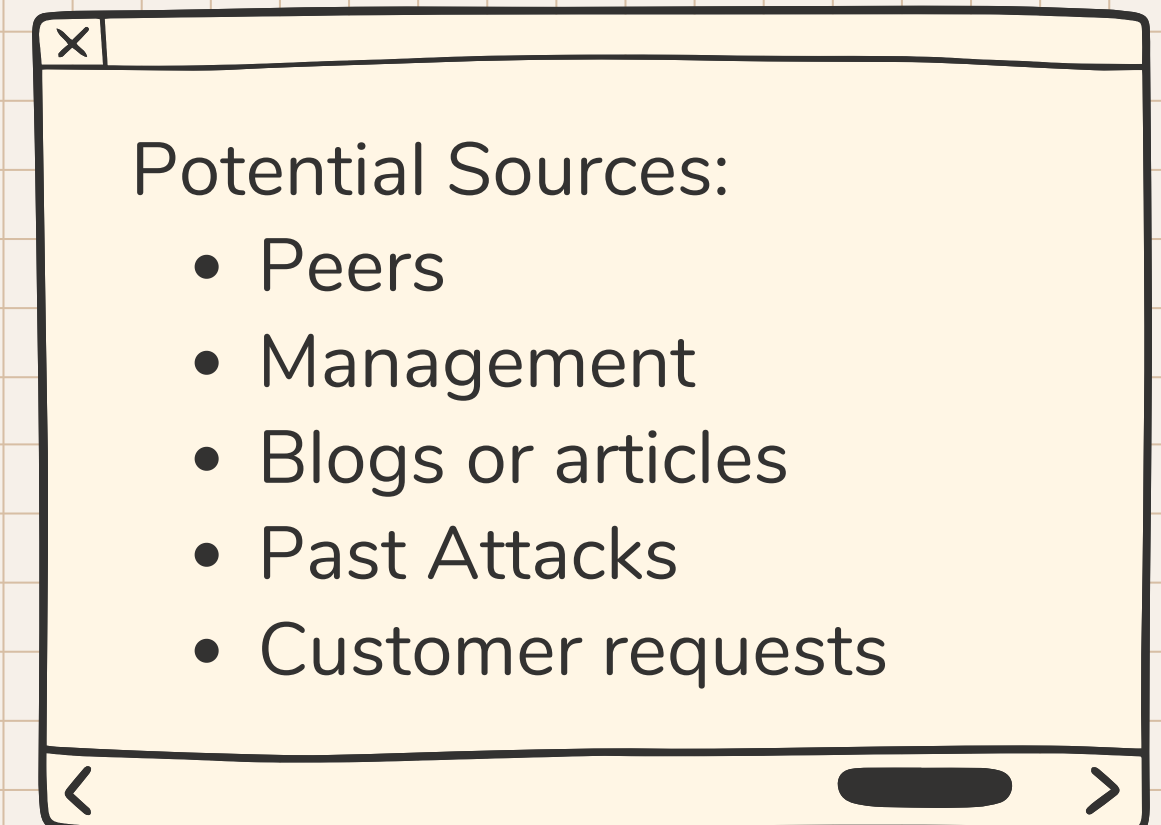
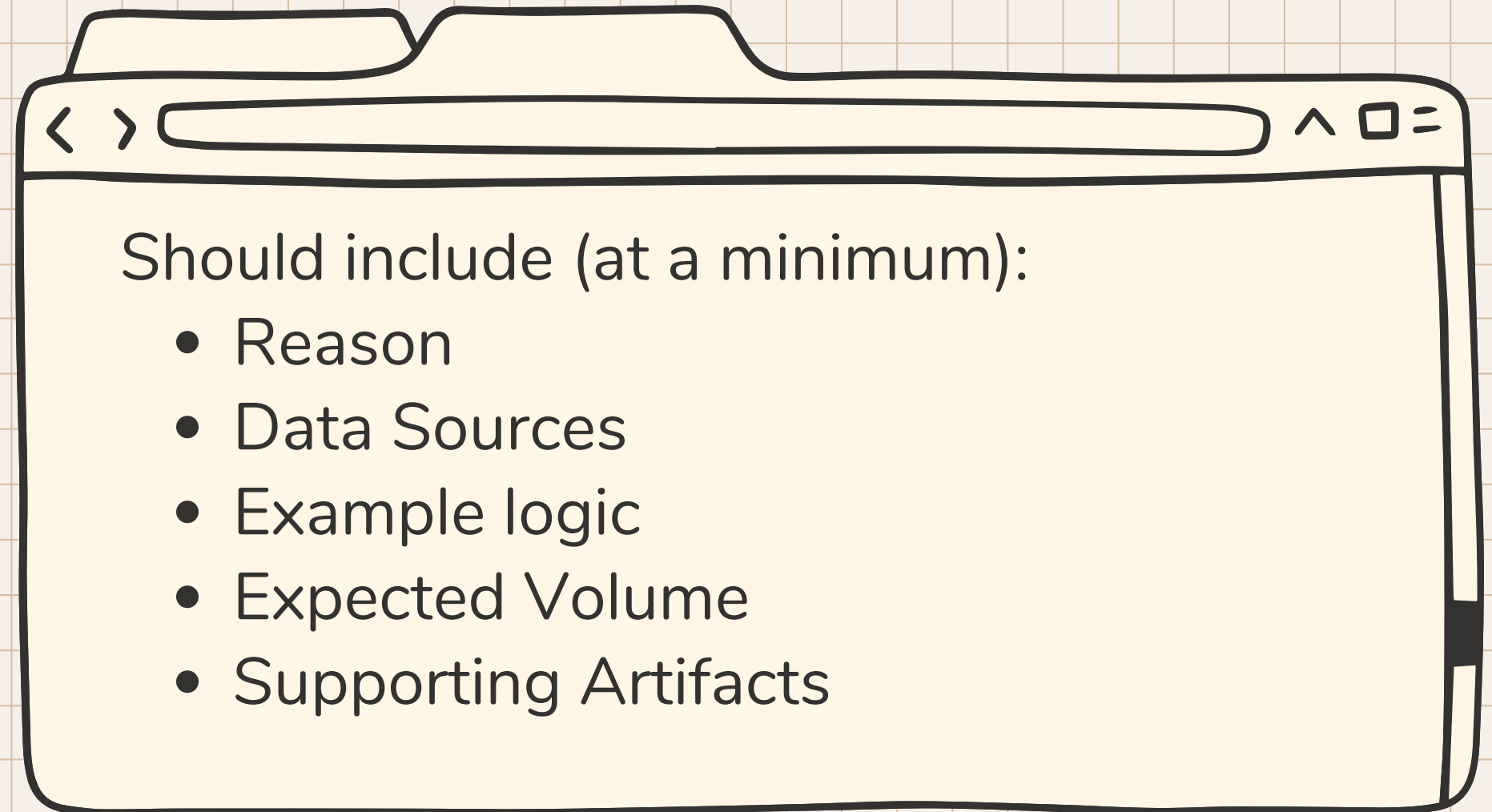
# The Scenario

PSEXEC plaintext authentication

# Step 0

# Detection Story

Collecting ideas in a way that sets your detection engineering team up for success is crucial



Step 0

# Detection Story

## Our Scenario

IOCs observed as part  
of malicious traffic

**k** process.executable

C:\[REDACTED]\PsExec.exe

**k** process.command\_line

PsExec.exe -accepteula @C:[REDACTED].txt -u  
"[REDACTED]\[REDACTED]" -p "[REDACTED]" cmd /c COPY "[REDACTED]  
[REDACTED].dll" "C:\Users\Public\"



Step 0

# Detection Story

## Our Scenario

IOCs observed as part  
of malicious traffic

*File with list of  
hostnames*

*Command to run  
after this point*

**k** process.executable

C:\[redacted]\PsExec.exe

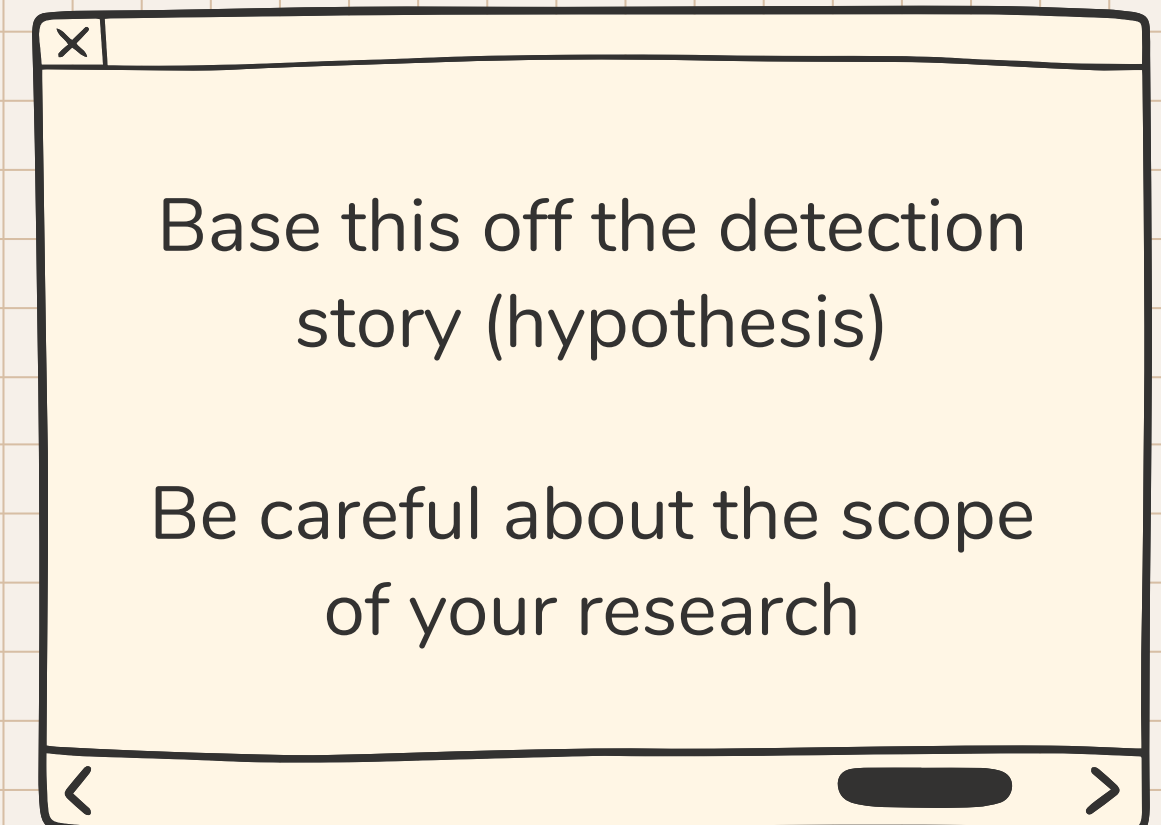
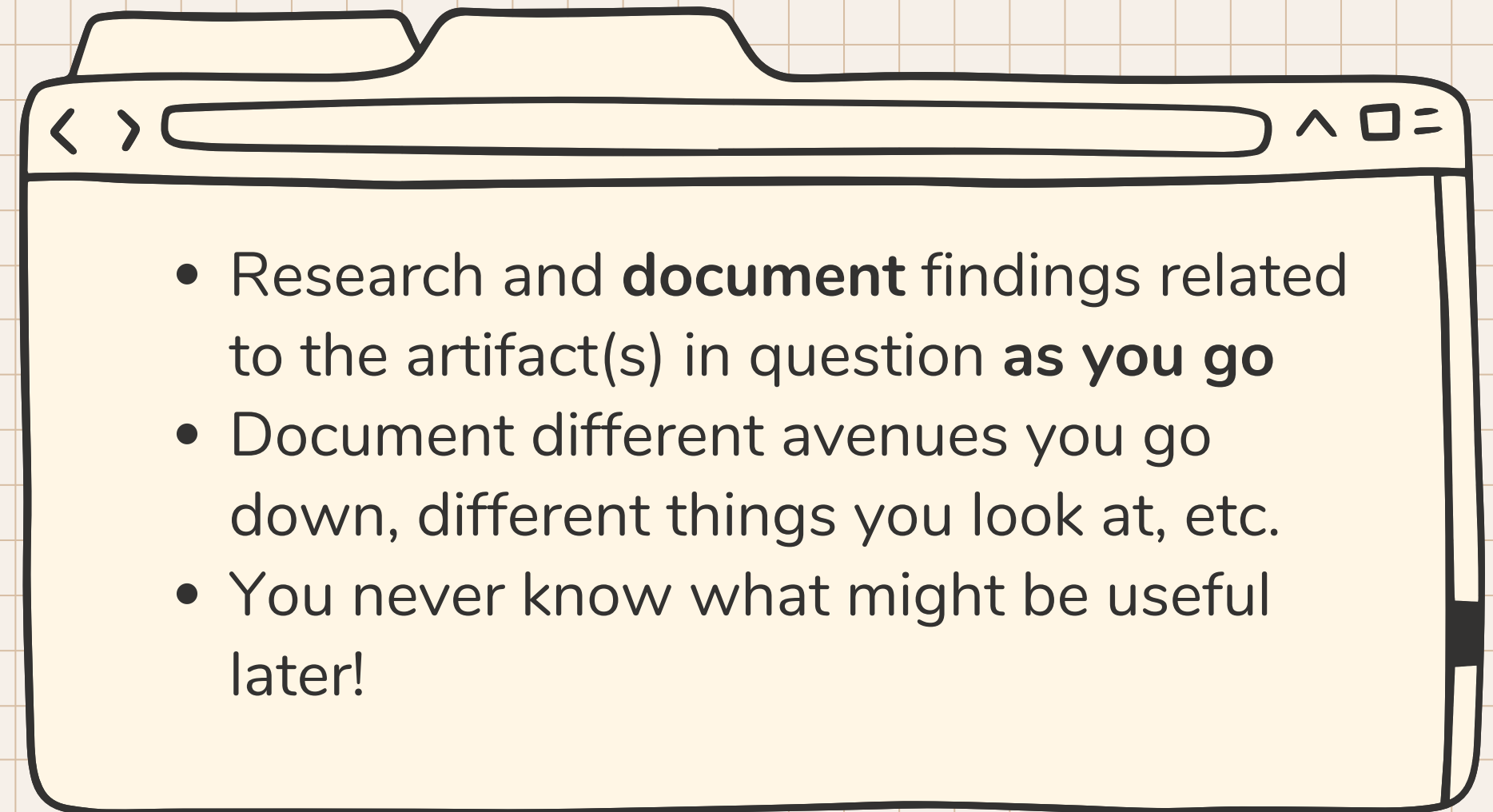
**k** process.command\_line

PsExec.exe -accepteula @C:\[redacted]list of hosts.txt -u  
"[redacted]Domain\ [redacted]User" -p "[redacted]password" cmd /c COPY "[redacted]  
[redacted]malware filepath [redacted].dll" "C:\Users\Public\"

# Step 1

## Research

A good detection requires full understanding of the idea



# Step 1

## Research

### Our Scenario

An understanding of the PSEXEC flags, and what the best practices are for plaintext auth

<https://learn.microsoft.com/en-us/sysinternals/downloads/psexec>

-p	Specifies optional password for user name. If you omit this you will be prompted to enter a hidden password.
-r	Specifies the name of the remote service to create or interact with.
-s	Run the remote process in the System account.
-u	Specifies optional user name for login to remote computer.
-v	Copy the specified file only if it has a higher version number or is newer on than the one on the remote system.
-w	Set the working directory of the process (relative to remote computer).

-p	Specifies optional password for user name. If you omit this you will be prompted to enter a hidden password.
----	--

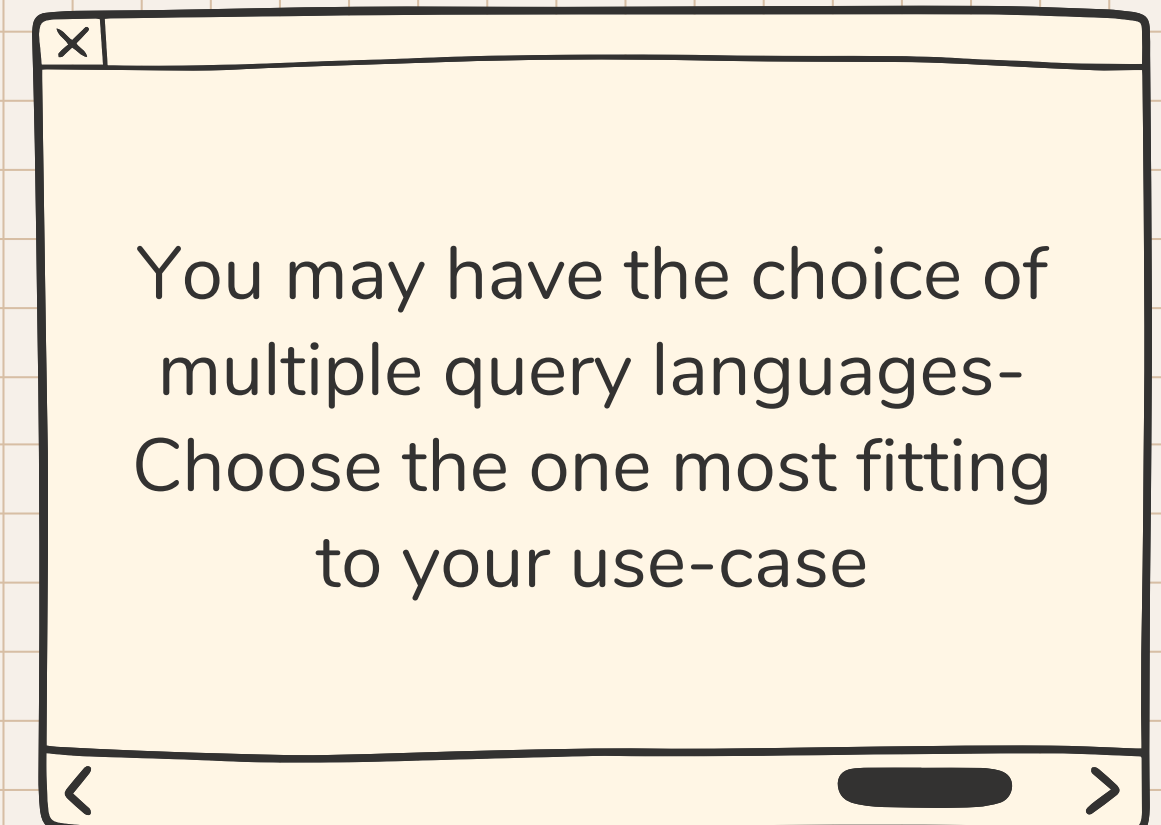
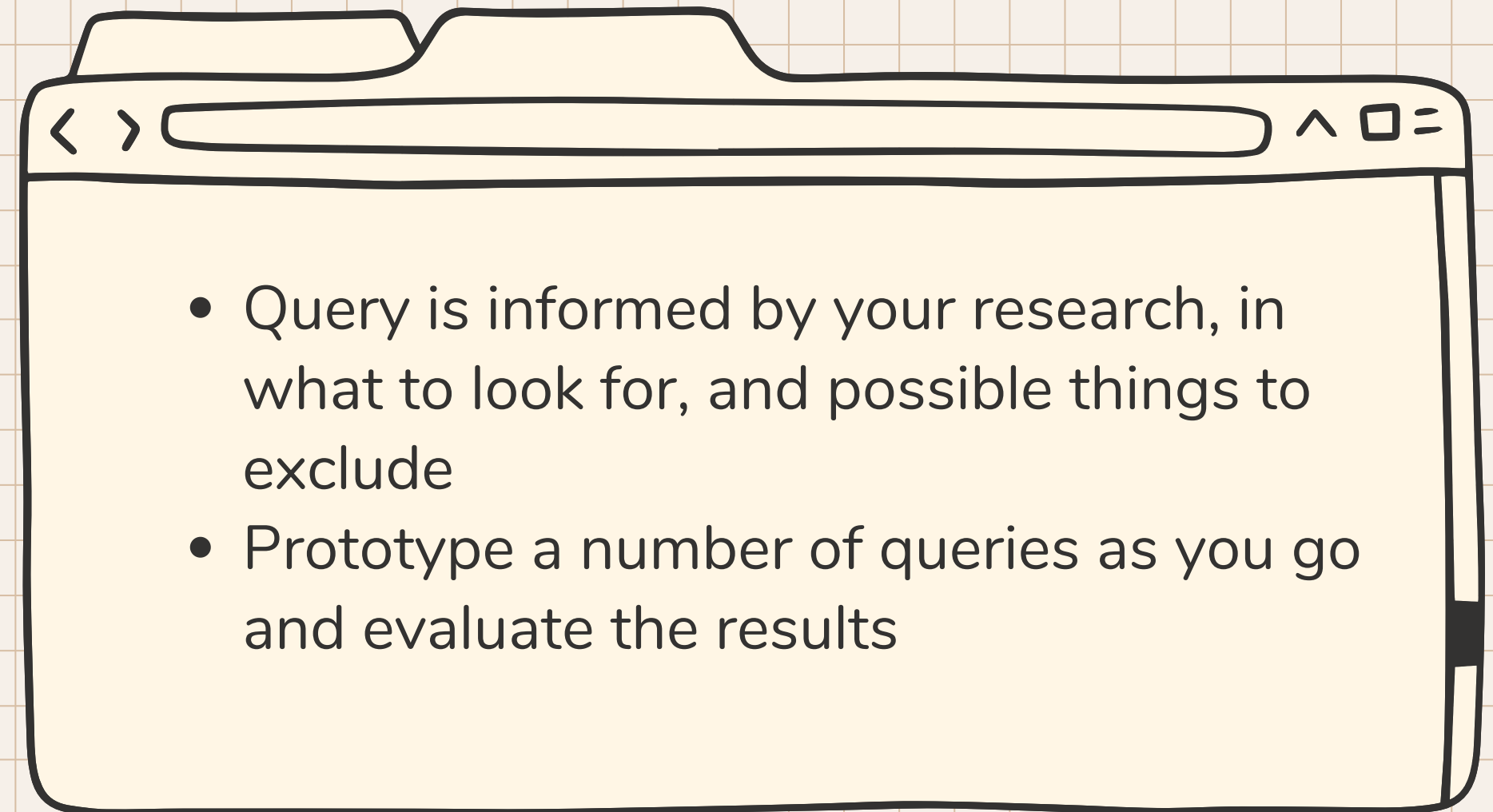


PSEXEC best practices

# Step 2

## Query

Without a good query, you don't have a detection.



# Finding a Balance

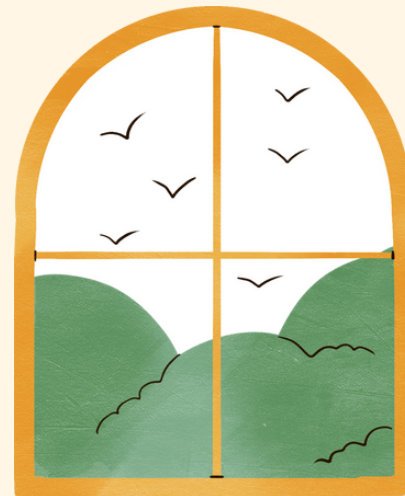
## Too broad

Risk burnout and high volume, missing key events buried in the pile



## Middle ground

Balance between volume and fidelity



## Too narrow

Specific to exact commands, any small change and you miss the event



# Step 2

## Query

### Our Scenario

A fairly simple KQL query is available to us, looking for the -p flag in PSEXEC executions

process.name: "PsExec.exe" and process.args: "-p"

process.name

Field	Value
k process.name	PsExec.exe

process.args

Field	Value
k process.args	[PsExec.exe, -accepteula, @C:\[REDACTED].txt, -u, [REDACTED], [REDACTED], -p [REDACTED], cmd, /c, COPY, \\\[REDACTED].dll, C:\Users\Public\c"]

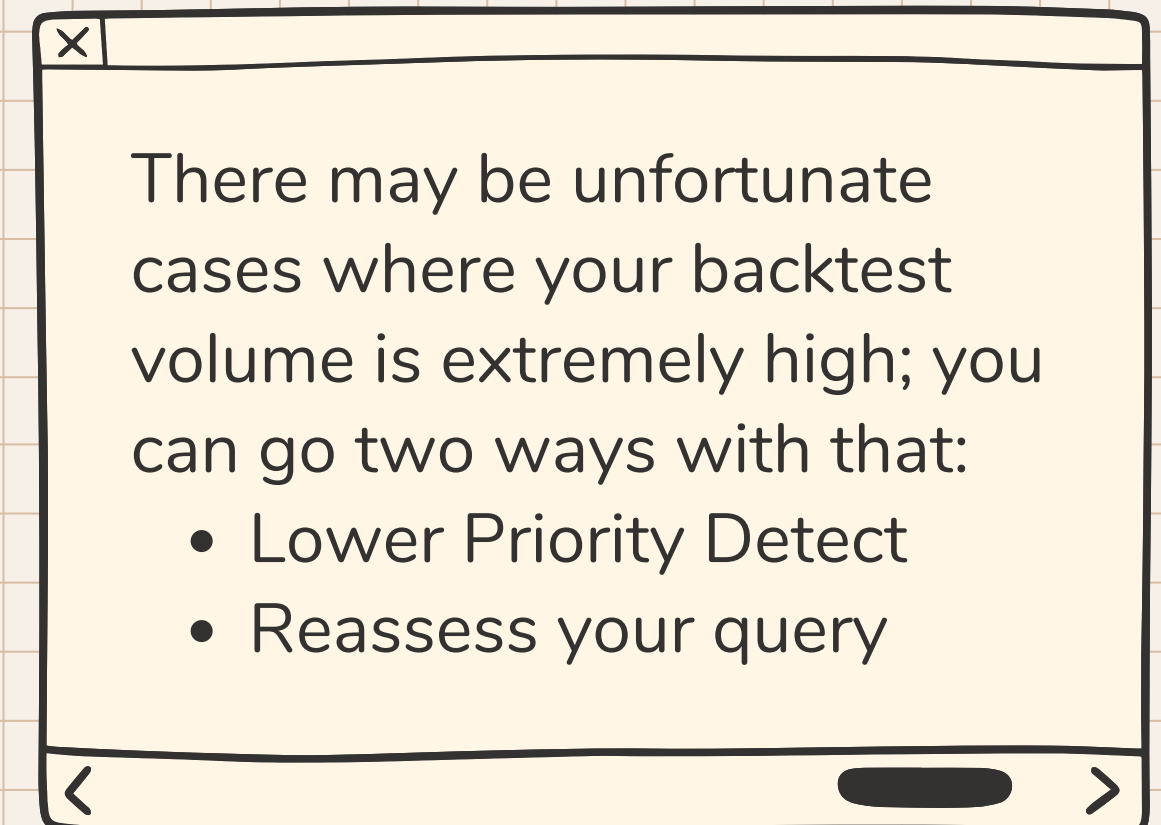
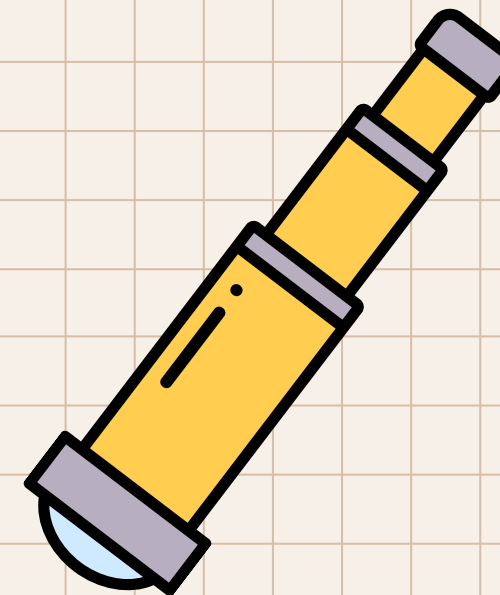
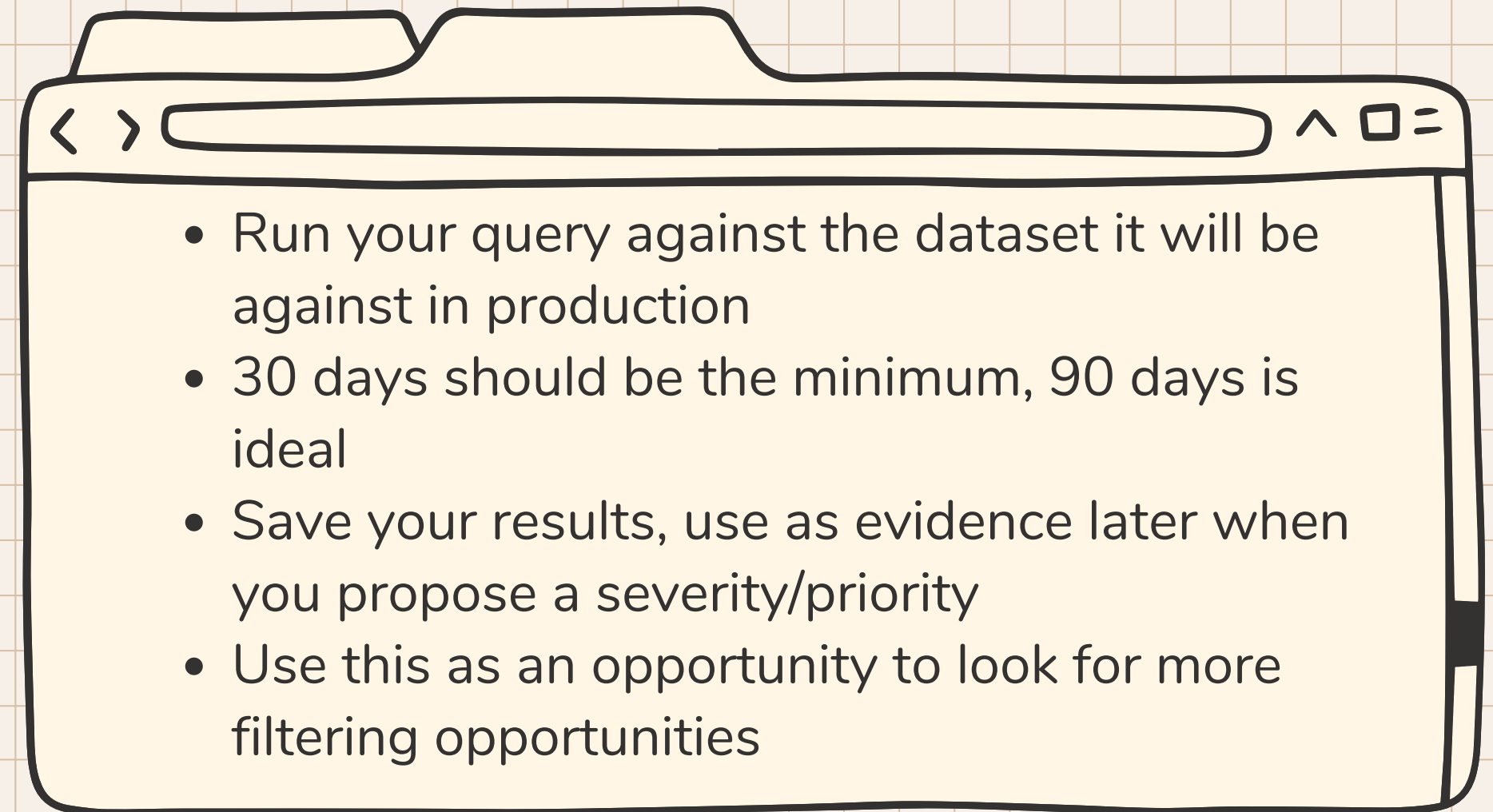
In this case you might want to consider lowering the process name, and finding PSEXEC knockoff names

Or better yet- looping in the PSEXEC service installation

# Step 3

## Backtest

Estimating the volume is a good thing to do \*before\* your SOC comes to you with pitchforks in hand



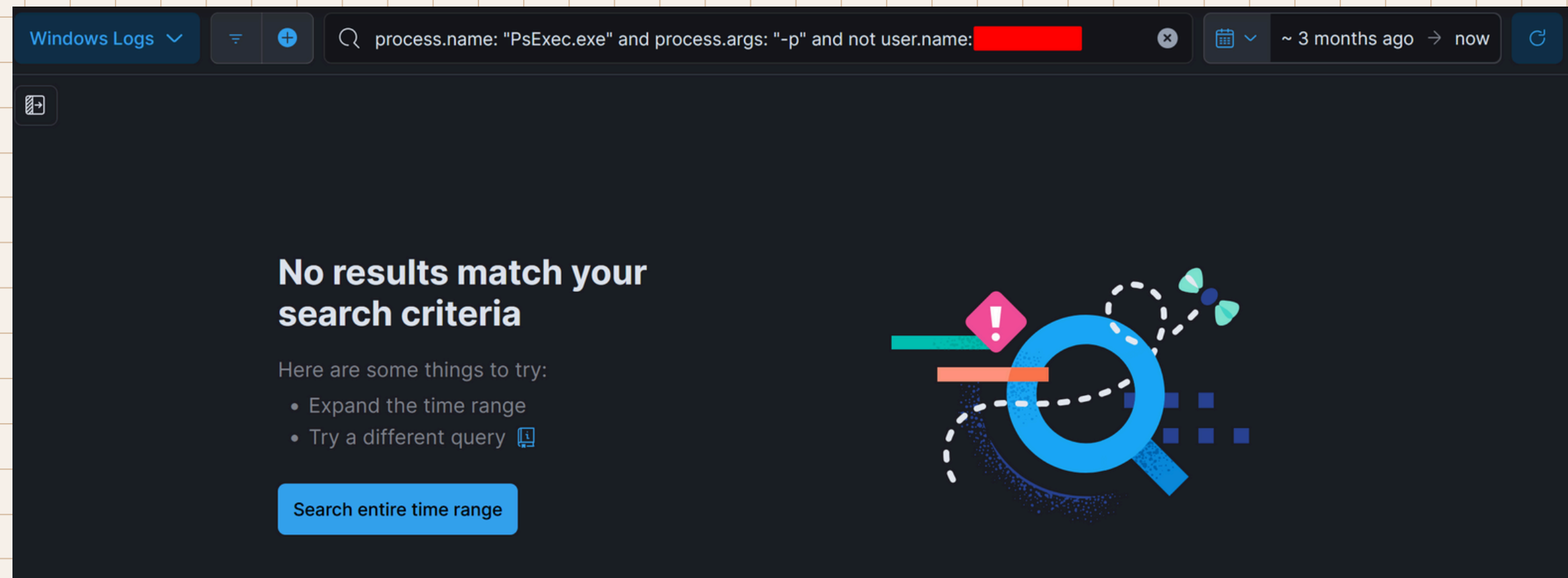


# Step 3

## Backtest

### Our Scenario

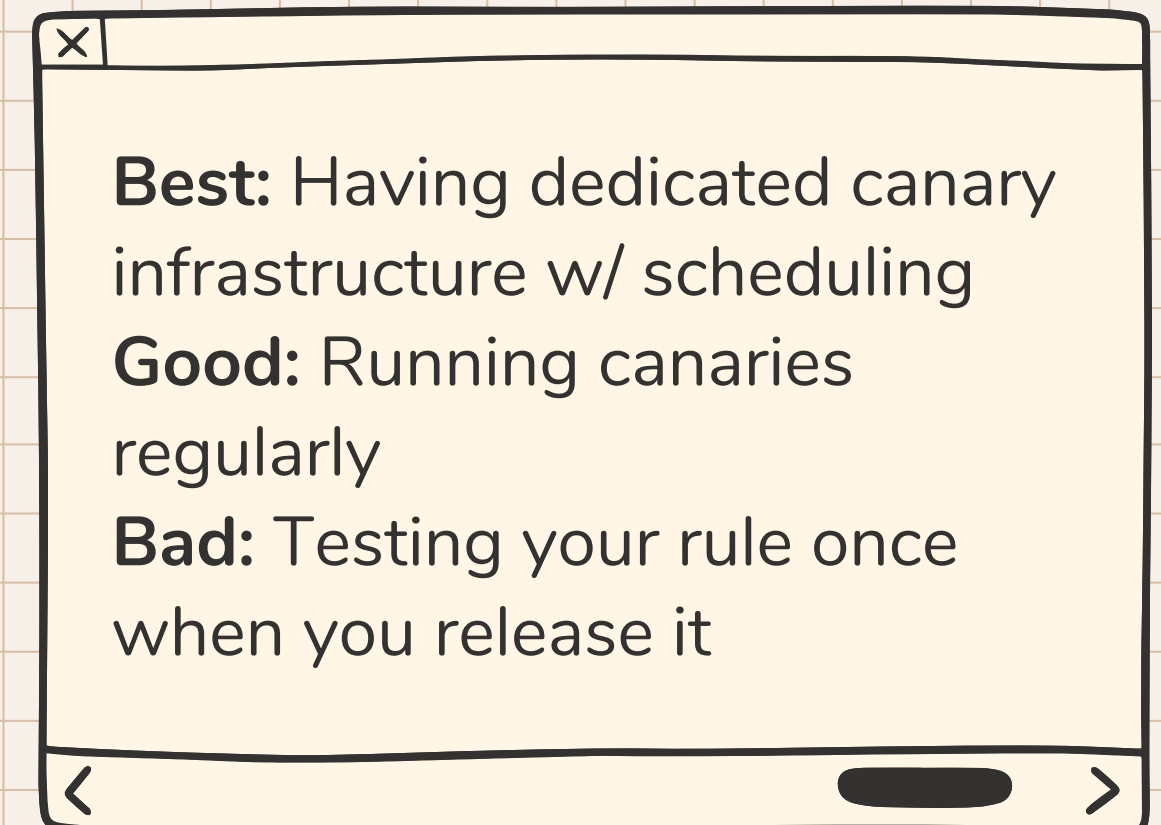
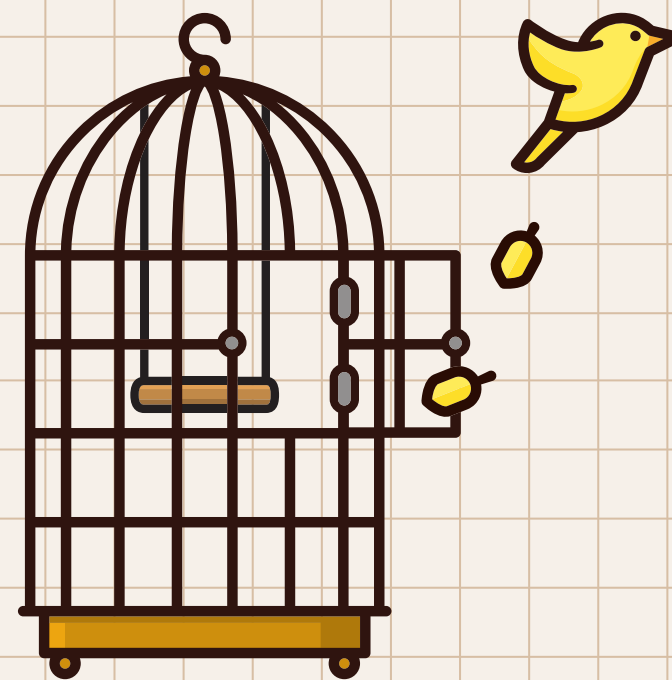
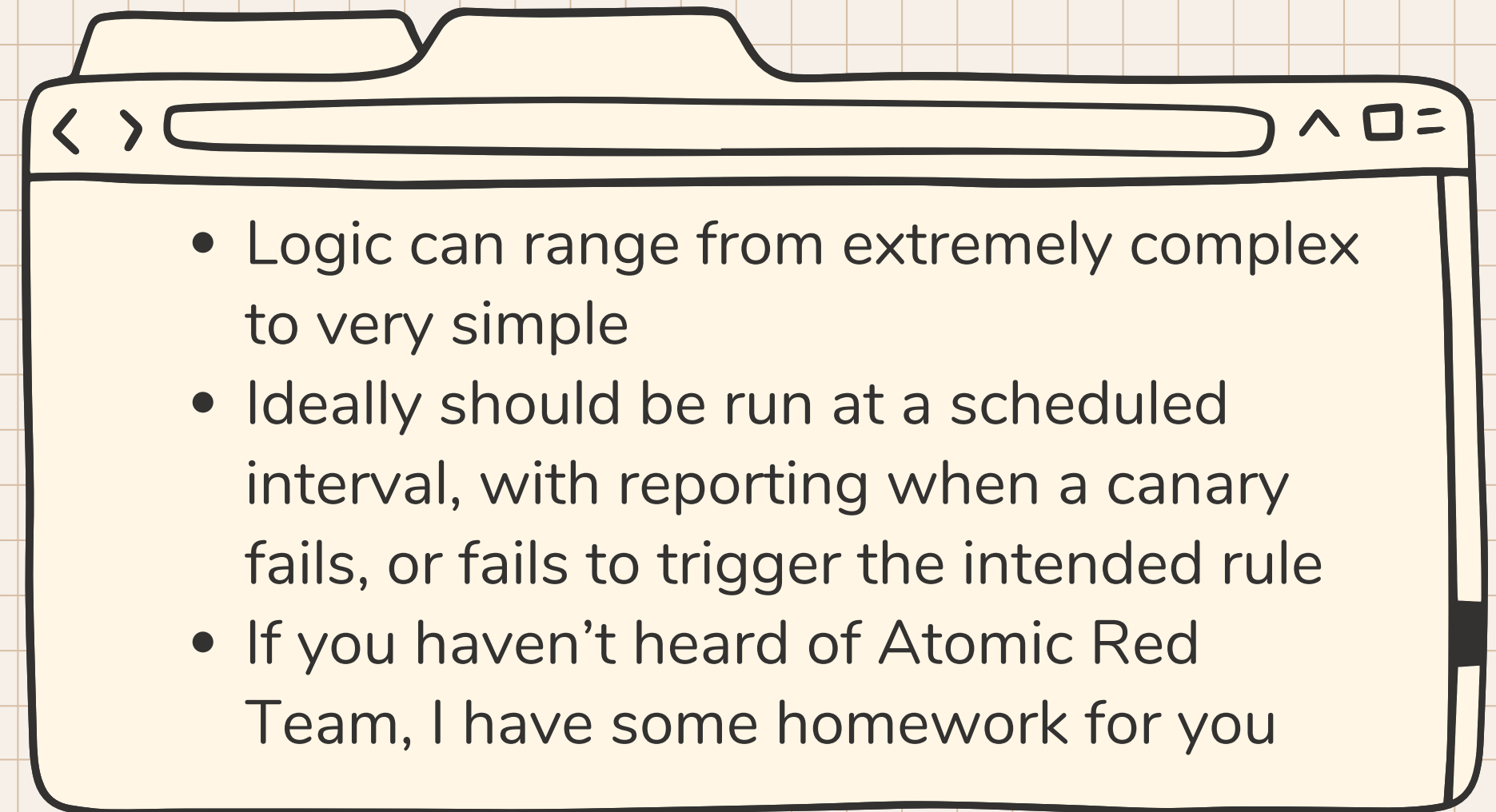
A 90-day backtest returned 0 unexpected results, which is excellent



# Step 4

## Canary

An important part of a detection, a canary can be run to ensure your detect still works as intended



## Step 4

# Canary

### Our Scenario

Very simple- We just have to run PSEXEC with the -p flag on a monitored host

```
psexec \\remote_computer -u username -p password command
```

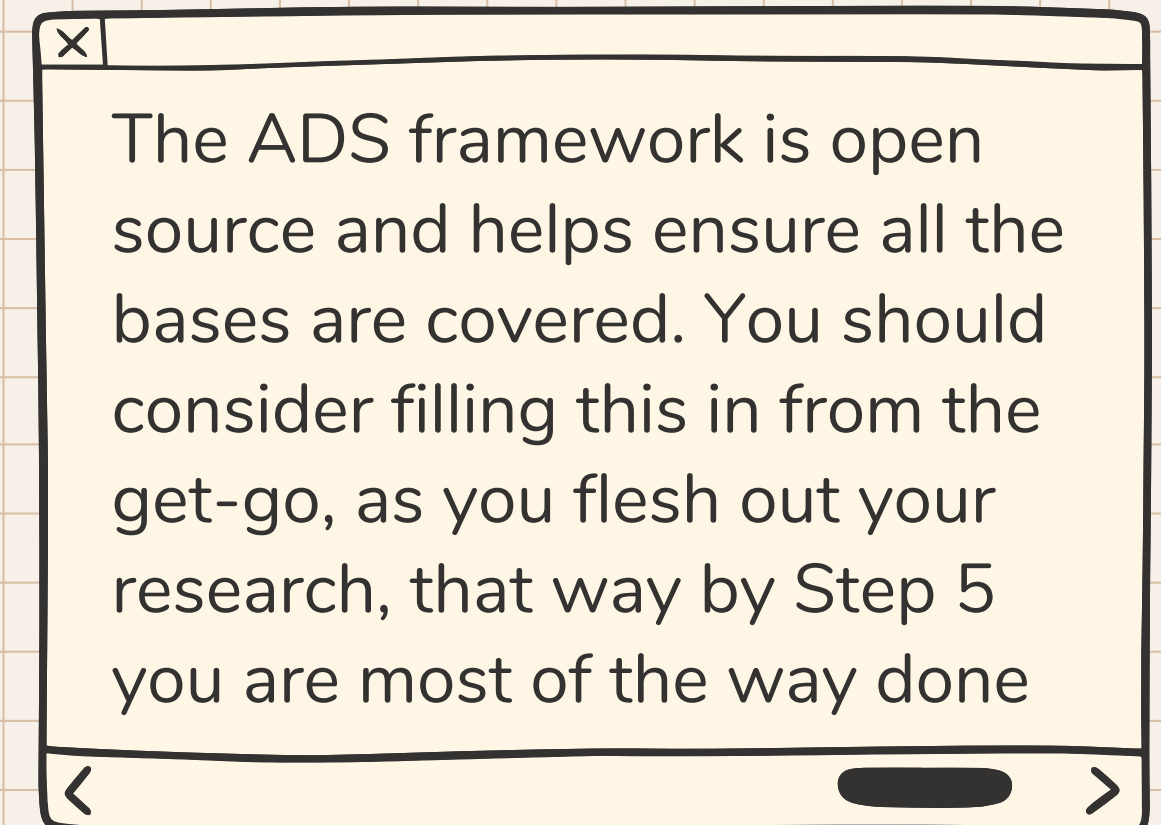
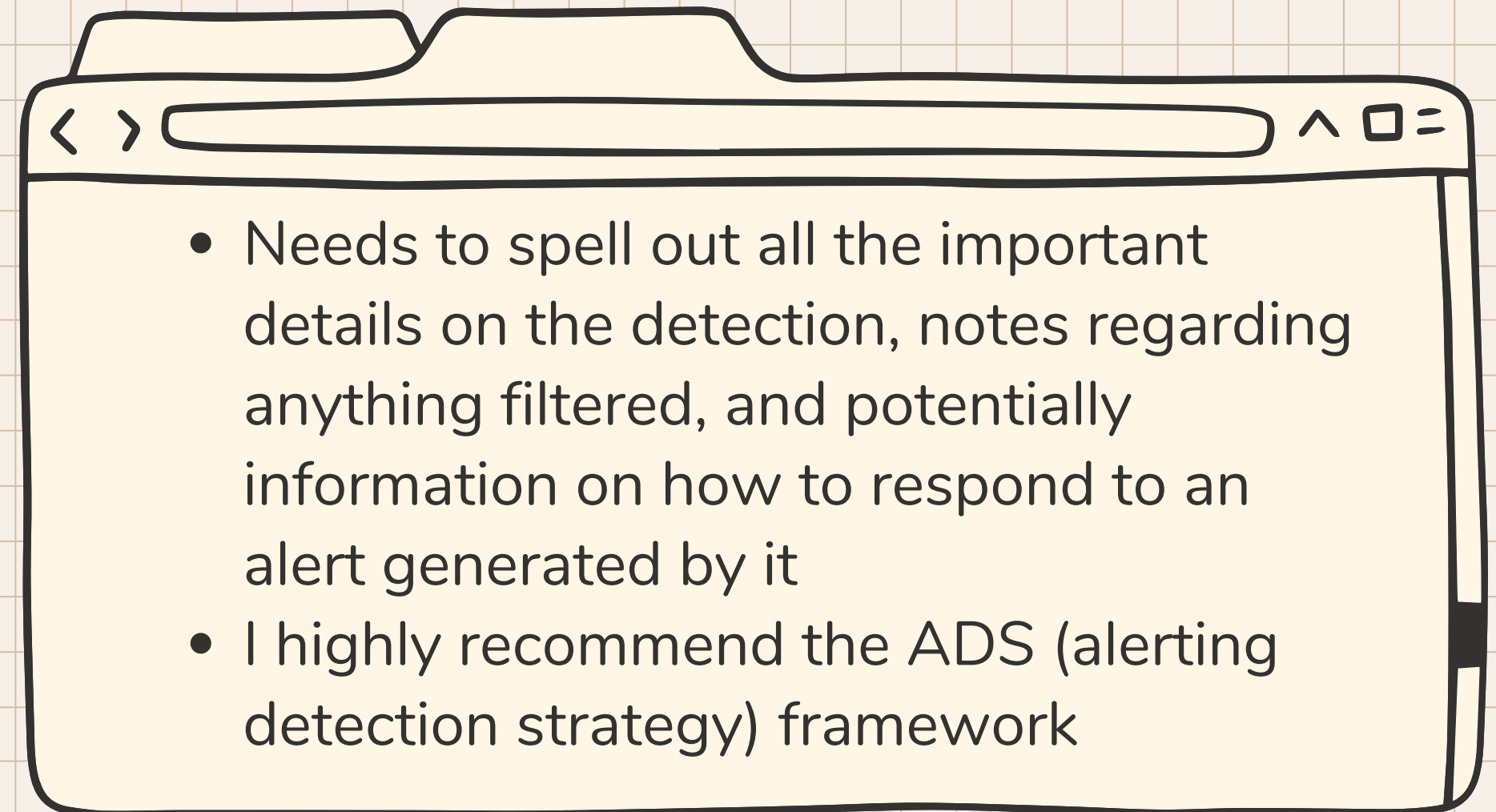
I told you... Simple

Don't overcomplicate it

# Step 5

## Documentation

I hear you groaning, but this is probably the most important step for a number of reasons



## Step 5

# Documentation

### Our Scenario

An ADS Doc on this detection is fairly straightforward, but hugely beneficial to both engineer and analyst

#### Goal

Detect the execution of `PsExec.exe` where a user password is passed over the commandline, which is against best practices and has been previously observed as an artifact of malicious activity.

#### Categorization

[Execution / Remote Service Execution](#)

[Lateral Movement / Remote Services](#)

#### Strategy Abstract

This detection looks for instances of `PsExec.exe` executing with the `-p` argument, which specifies a password authentication to the user specified, but over the commandline rather than in a protected session. This combination could indicate attacker lateral movement or execution.

#### Technical Context

Psexec is a legitimate Sysinternals tool frequently used by administrators to execute commands on remote systems. However, it is commonly abused by threat actors to perform lateral movement and execute commands across networks. The presence of `PsExec.exe` with the `-p` flag in command-line arguments was previously observed as a malicious indicator.

Detection logic:

```
process.name: "PsExec.exe" AND process.args: "-p"
```

#### Blind Spots and Assumptions

- **Blind Spots:** Renamed or obfuscated `PsExec.exe` binaries
- **Assumptions:** Assumes the attacker will use `PsExec` with the `-p` flag instead of following best practice

#### False Positives

# Step 5

## Documentation

Huge shoutout to Wade Wells for showing me this WWHF!

I actually used it to generate the first draft of the ADS doc on the last slide

## BONUS

Write me an ADS for this detection logic: process.name: "PsExec.exe" and process.args: "-p"



### Goal

Detect the execution of `PsExec.exe` with a password-protected session, which is indicative of potential remote code execution attempts or lateral movement in the network.

### Categorization

Execution / Remote Service Execution

Lateral Movement / Remote Services

### Strategy Abstract

This detection looks for instances of `PsExec.exe` executing with the `-p` argument, which specifies a password-protected session. Monitoring this combination aims to identify attempts to execute commands on remote machines, typically associated with lateral movement or initial infection spread.

### Technical Context

PsExec is a legitimate Sysinternals tool frequently used by administrators to execute commands on remote systems. However, it is commonly abused by threat actors to perform lateral movement and execute commands stealthily across networks. The presence of `PsExec.exe` with the `-p` flag in command-line arguments ( `process.args` ) is specifically notable as it indicates credential usage for remote execution, a tactic used to bypass authentication prompts and gain seamless access.

- **Process Name:** `PsExec.exe`
- **Process Argument:** `-p` (indicates a password-based session)

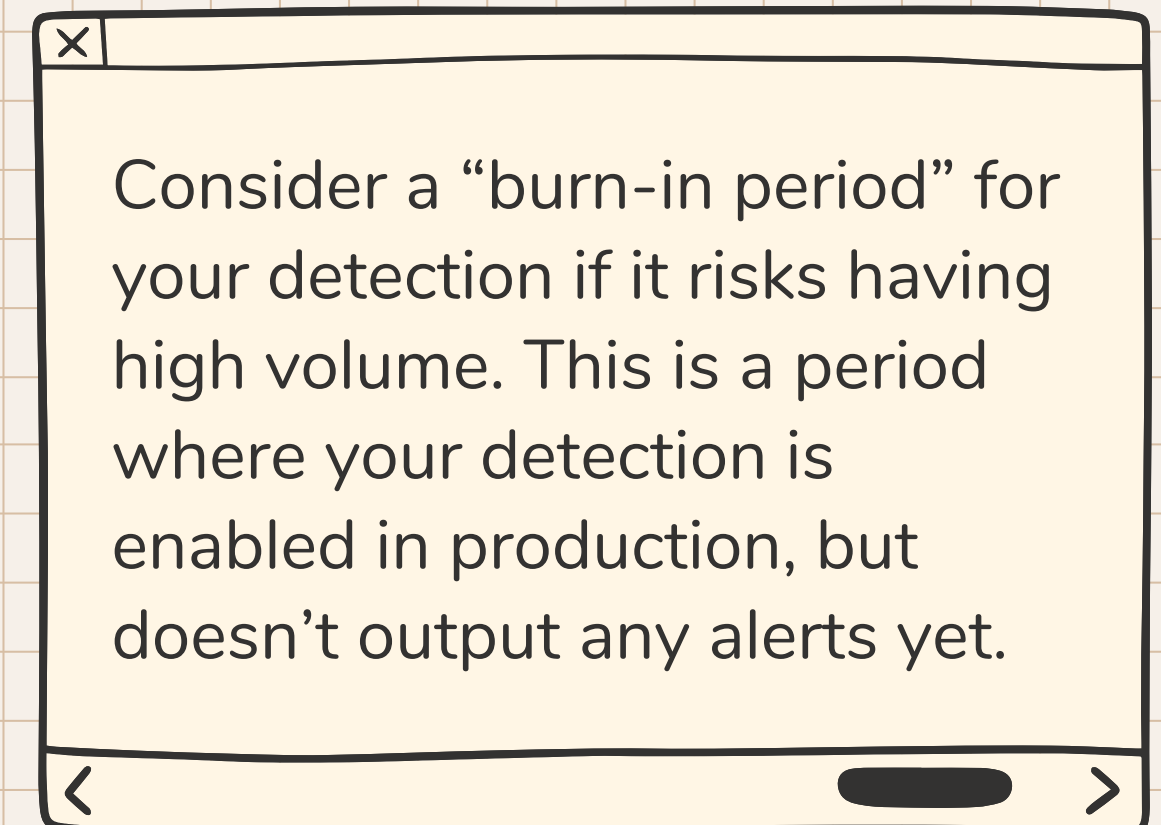
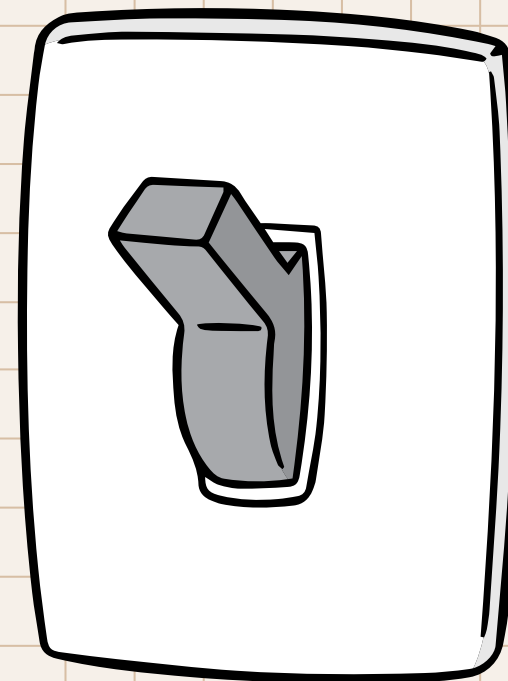
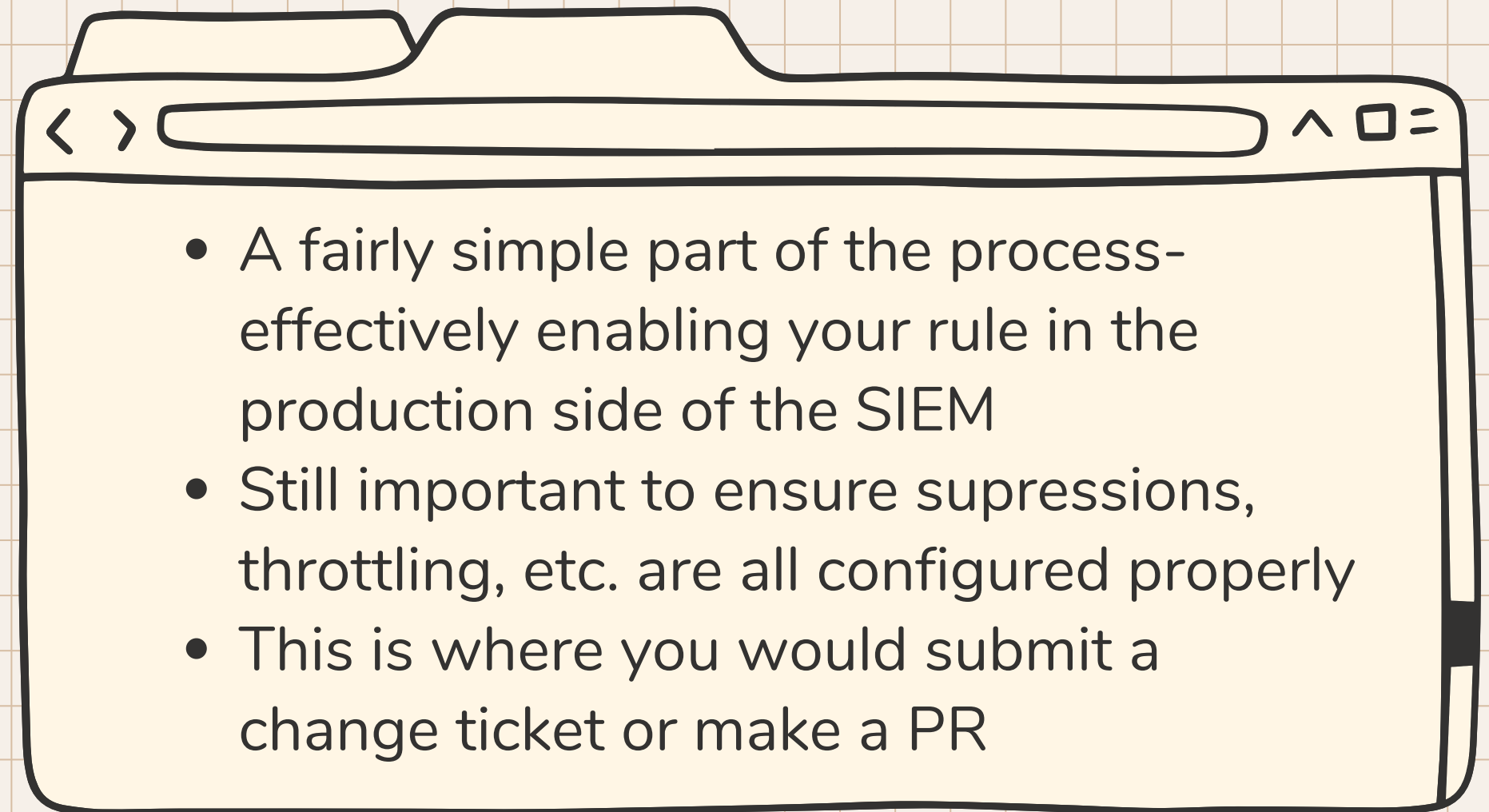
<https://chatgpt.com/g/g-VZjS4zo1C-ads-framework-bot>



# Step 6

## Onboarding

This is where you see return for your efforts, by onboarding your detection into the SIEM





# Step 6

## Onboarding

### Our Scenario

Our detection has a lengthy backtest with 0 results- so for us it's just about configuring the rule

**Source**  
Use Kibana [Data Views](#) or specify individual [index patterns](#) as your rule's data source to be searched.

☒ **Index Patterns** ☐ **Data View**

**Index patterns** [Reset to default index patterns](#)

Enter the pattern of Elasticsearch indices where you would like this rule to run. By default, these will include index patterns defined in Security Solution advanced settings.

**Custom query** [Import query from saved timeline](#)

**Suppress alerts by** Optional

Select field(s) to use for suppressing extra alerts

☒ **Per rule execution**

☐ **Per time period**

**If a suppression field is missing** [?](#)

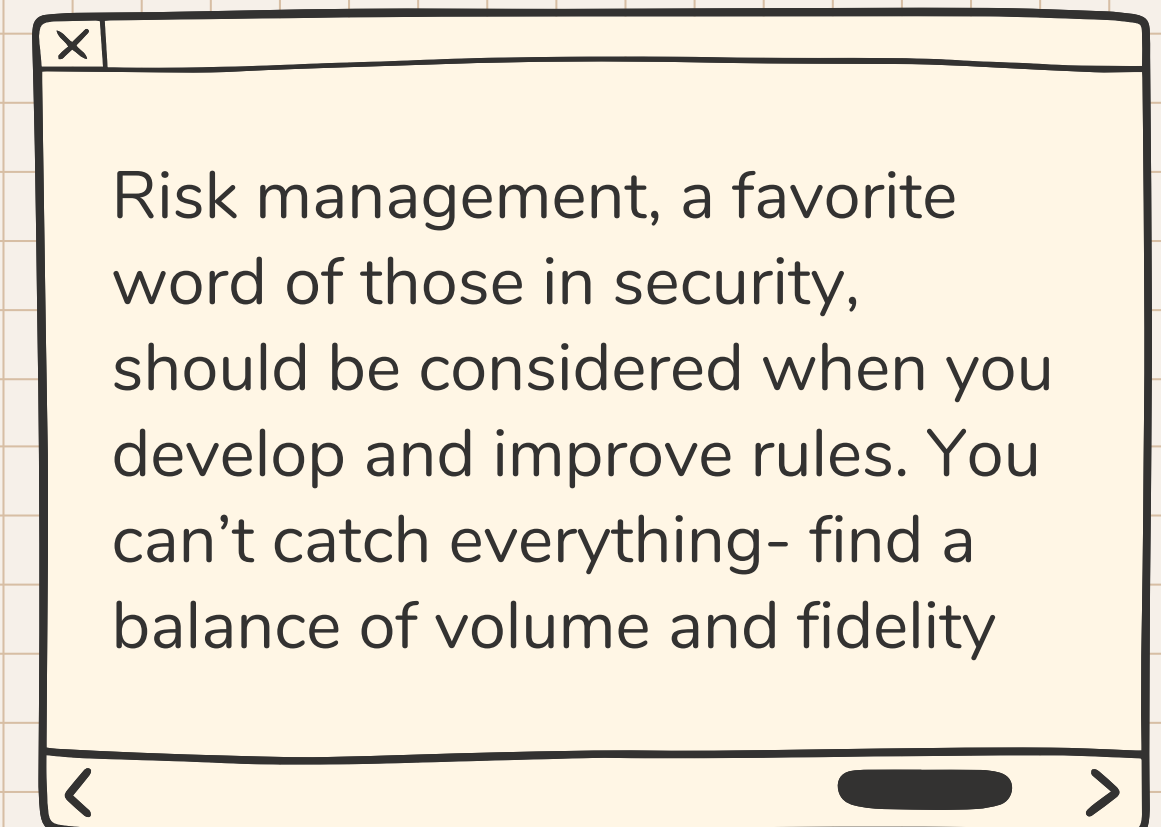
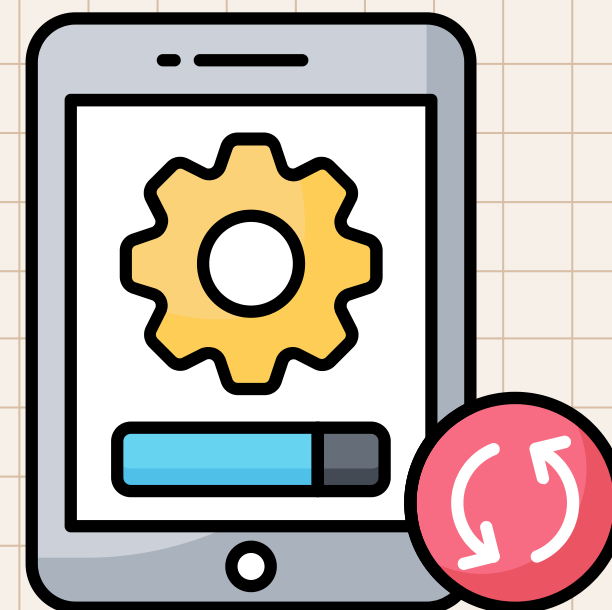
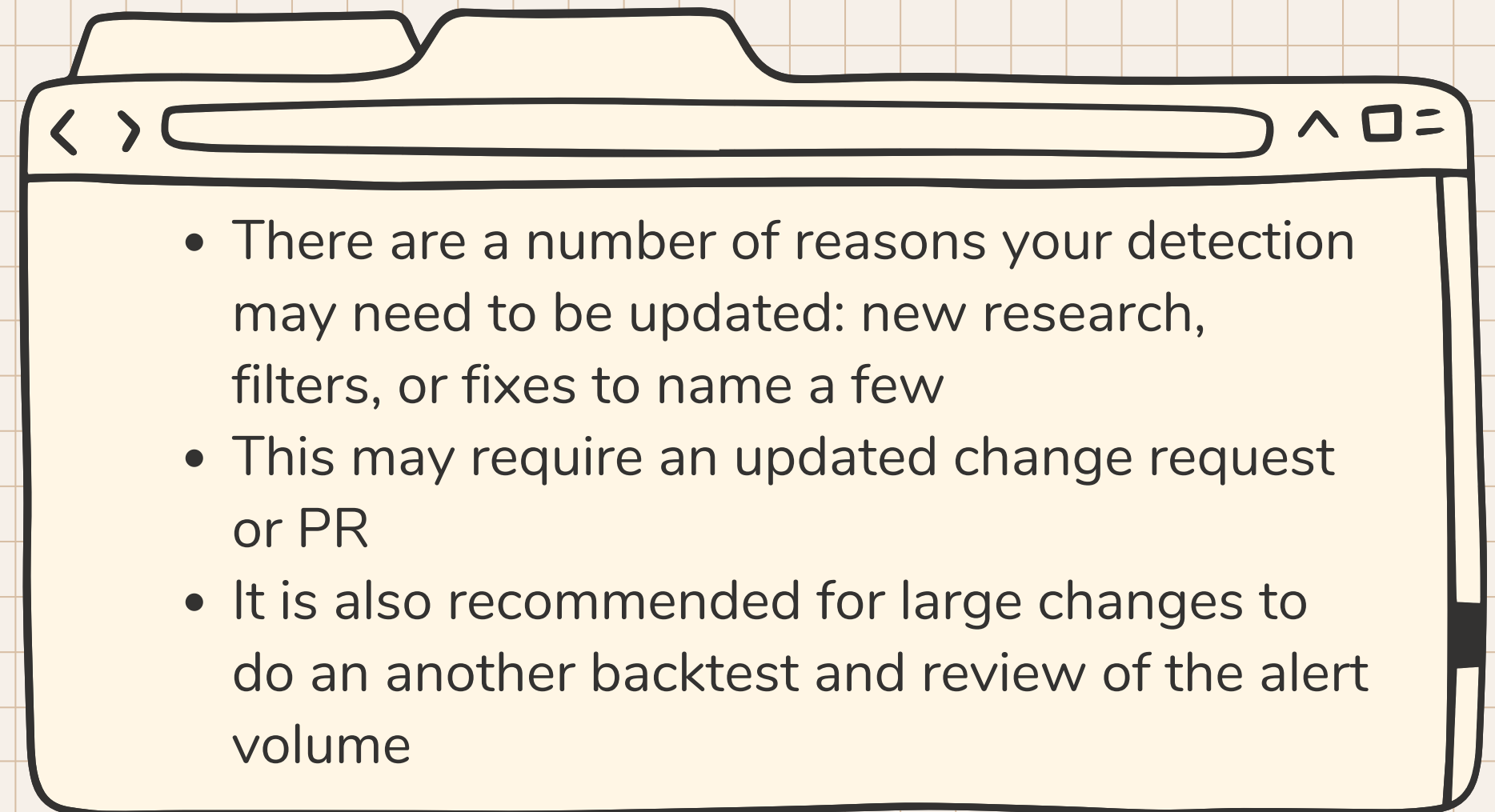
☐ **Suppress and group alerts for events with missing fields**

☒ **Do not suppress alerts for events with missing fields**

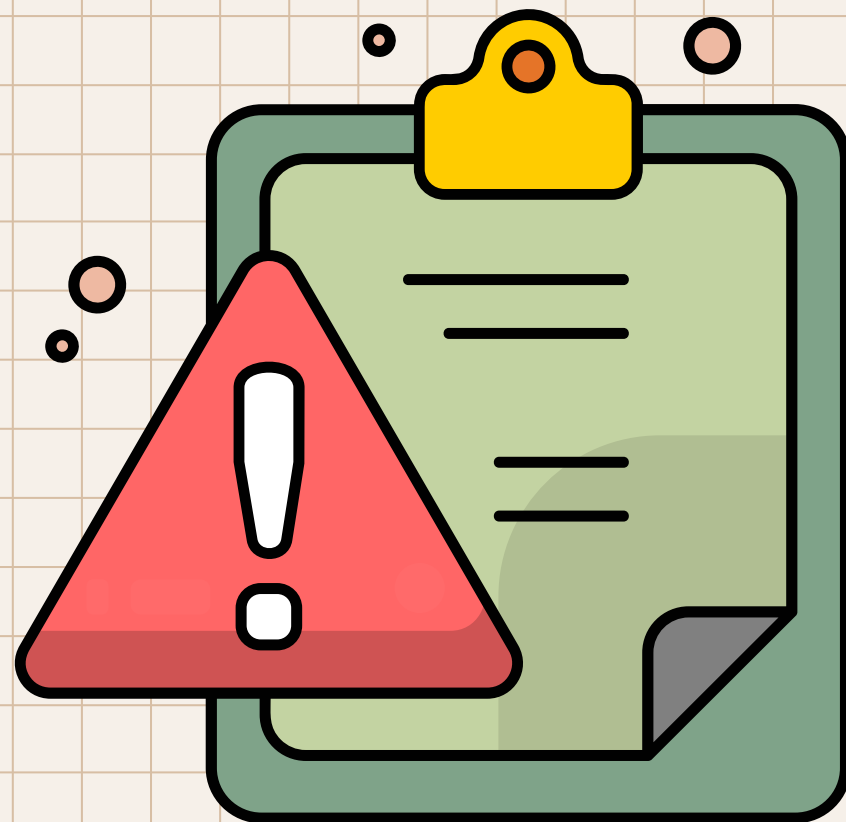
# Step 7

## Continuous Improvement

Detection engineering doesn't stop when you hit approve or enable, it should be a continuous process



# Risk



## Filters & Risk

There will almost always be risk involved when you filter activity out and develop detections- You have to manage and balance that risk just like with everything else

### Avoidance

Filtering very selectively and dealing with more potential volume

### Transfer

Scheduling threat hunts to look for a certain TTP on a regular cadence

### Middle Ground

### Acceptance

Sometimes a high volume detection is not worth the value it brings

### Mitigation

Having other detections for similar or related TTPs (defense in depth)

## Step 7

# Continuous Improvement

### Our Scenario

Pretend we have an admin that needs to execute PSEXEC this way, and we accept that risk; filter them out if you can

### Add rule exception

Exception name

Admin User

#### Conditions

Alerts are generated when the rule's conditions are met, except when:

Field

Operator

Value

user.name

× ▼

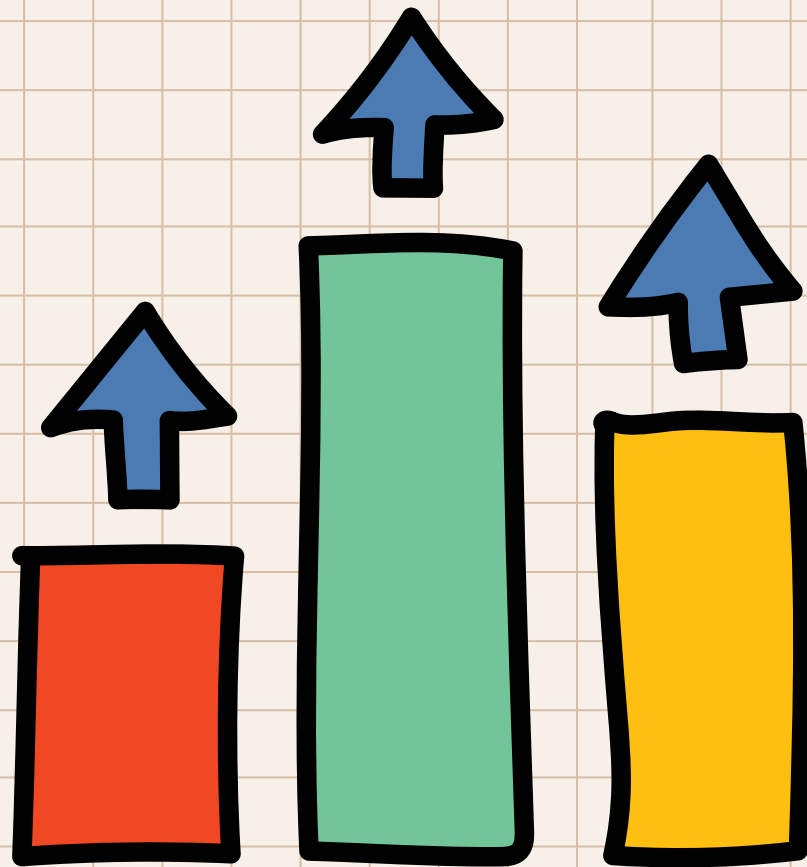
is

▼

hayden.covington

▼

# Continuous Improvement



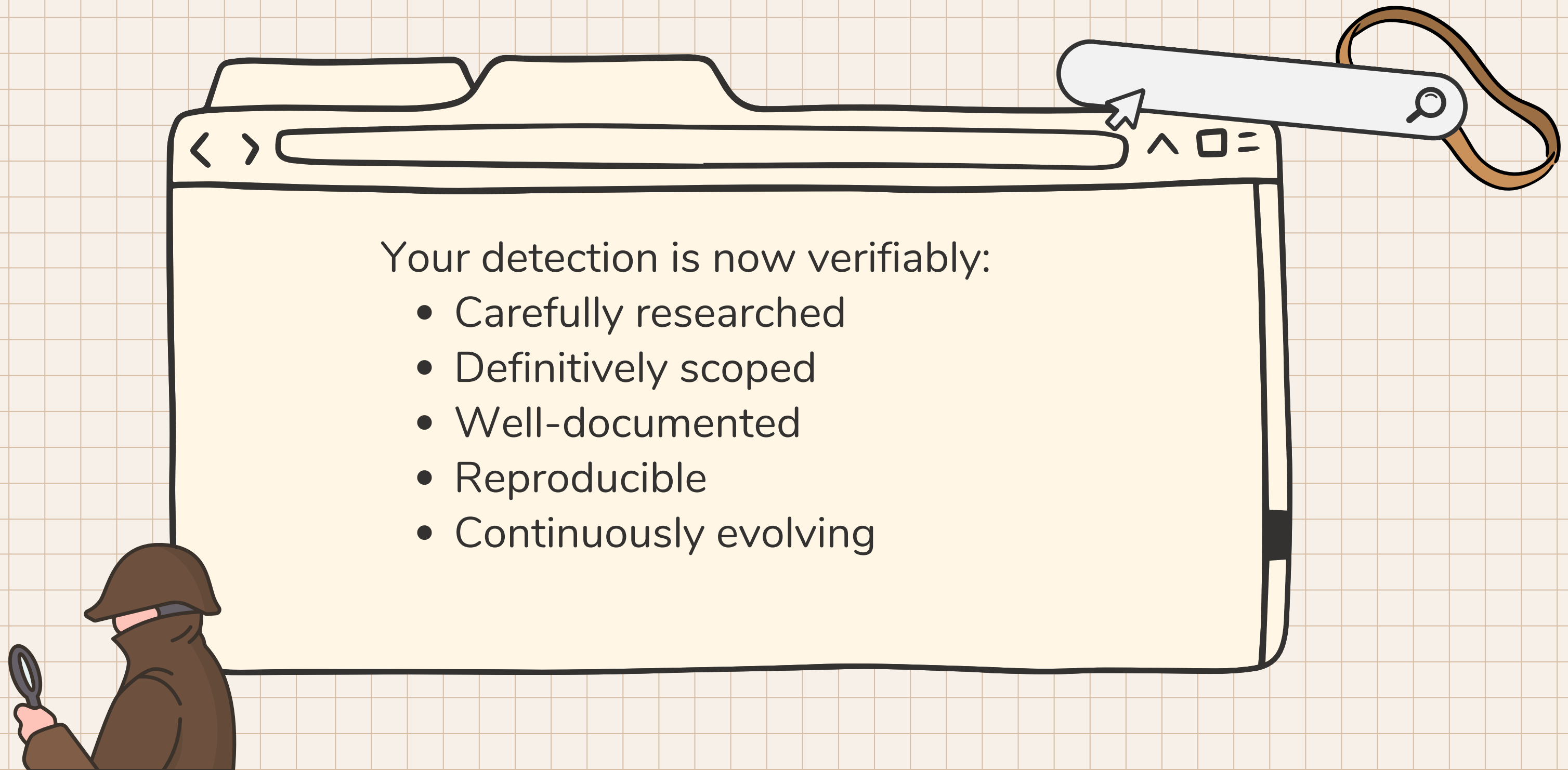
Changes can be brought about by:

- Operations
- New research
- etc etc etc

Can range from:

- Filters or suppressions
- Minor query changes
- Full Overhauls

# Detection = Theory





...

I told you there  
was a point.



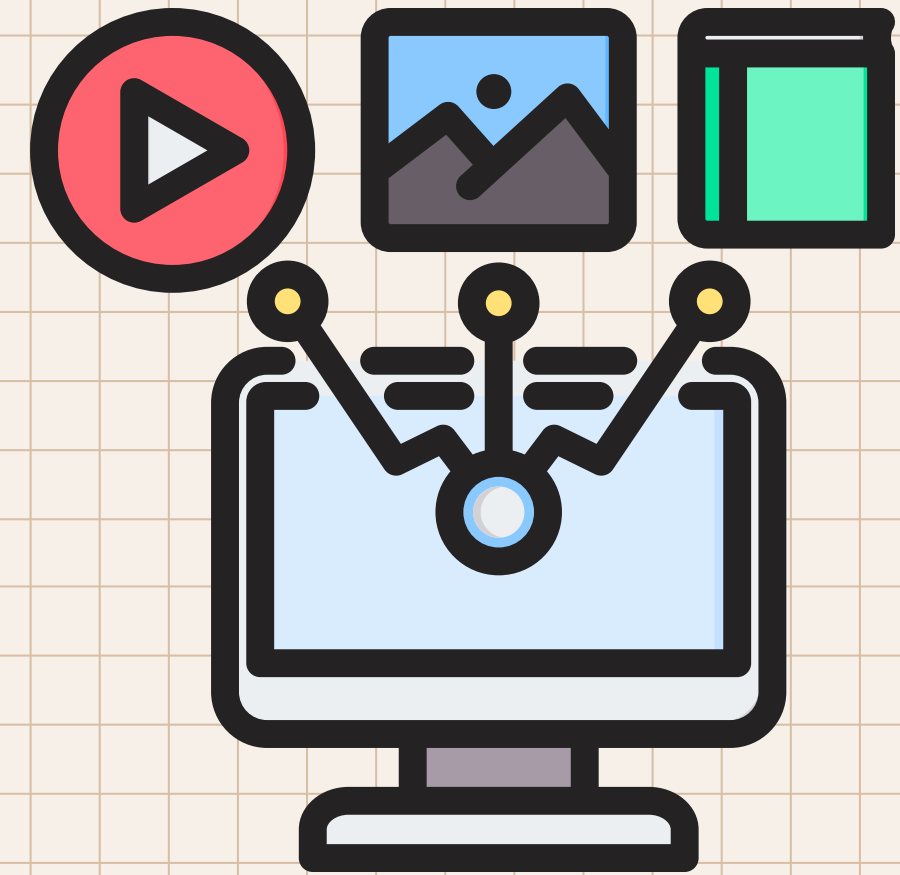
...

"I understand the  
connection now"  
- You, hopefully

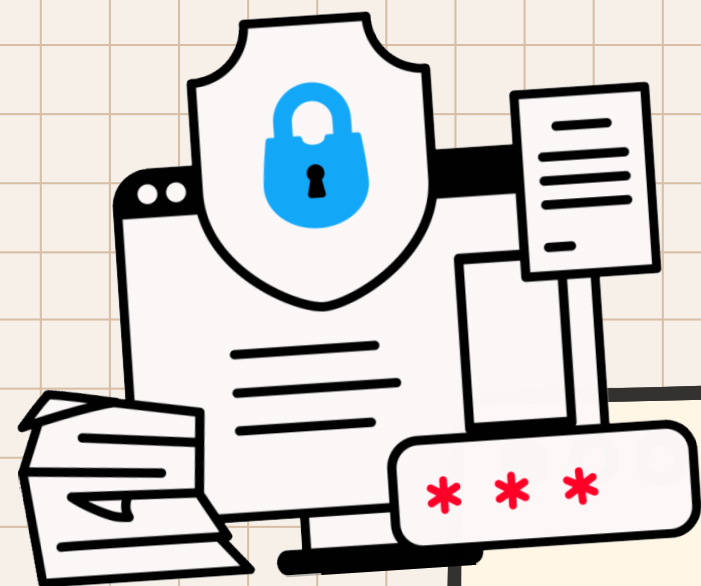


# Resources

- <https://github.com/palantir/alerting-detection-strategy-framework>
- <https://chatgpt.com/g/g-VZjS4zolC-ads-framework-bot>
- [https://www.splunk.com/en\\_us/blog/learn/detection-as-code.html](https://www.splunk.com/en_us/blog/learn/detection-as-code.html)

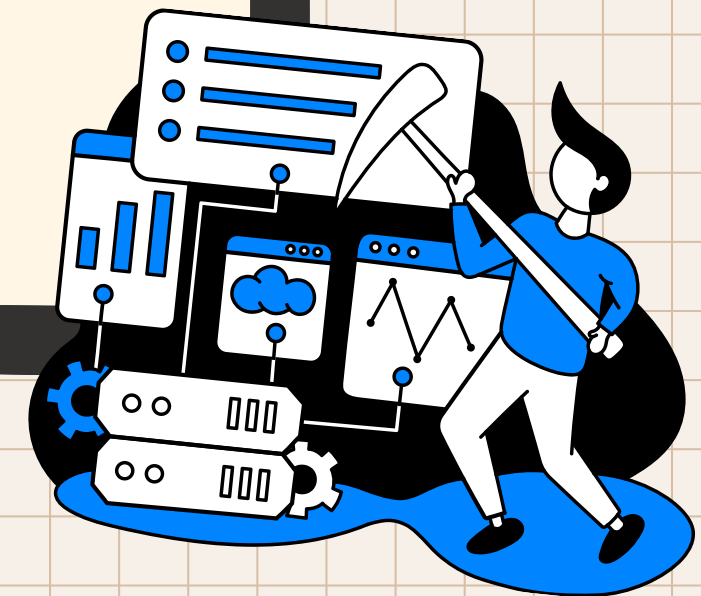
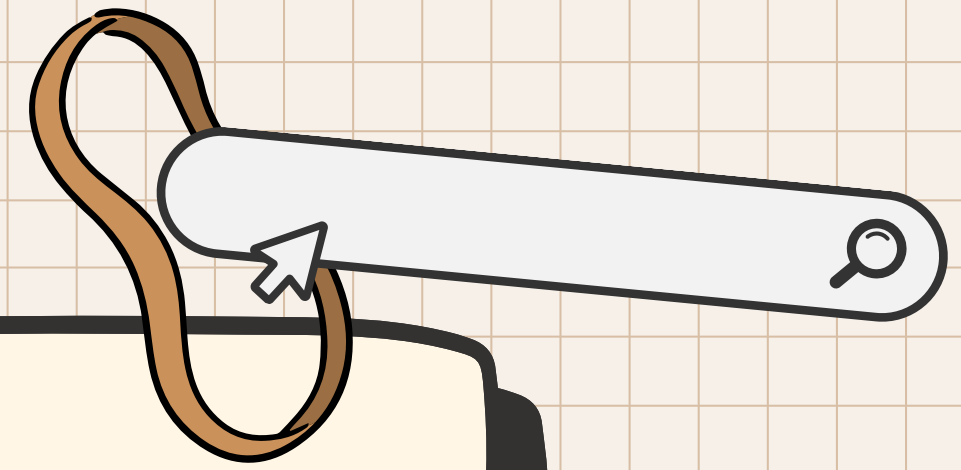


Shameless plug  
incoming



# Foundations of SOC with Elastic and Jira

Antisyphon Secure Code Summit in December  
and LIVE WWHF @ Mile High in February



<https://www.antisyphontraining.com/course/foundations-of-soc-with-elastic-and-jira-with-hayden-covington>



@KilobyteTheDust