TLP:CLEAR

We All Want Validation (in our detections)

John Stoner (Google Cloud, US)

#whoami > John Stoner

SIEM/SecOps for over 20 years

- Detection Engineering, Threat Hunting and Threat Intelligence

Focus on content development

Build adversary emulations (APT focus)

Blog on SecOps - New to Google SecOps

How We Got Here

Love emulations (and simulations)

- Build CTF scenarios
- Use these data sets for analyst training
  - Detection Engineering
  - Threat Hunting
  - Triage and Investigation

Pitfalls associated with building detections in these emulated (static) environment

How can we better serve detection engineers?

Not A One Size Fits All Challenge

Loads of moving parts

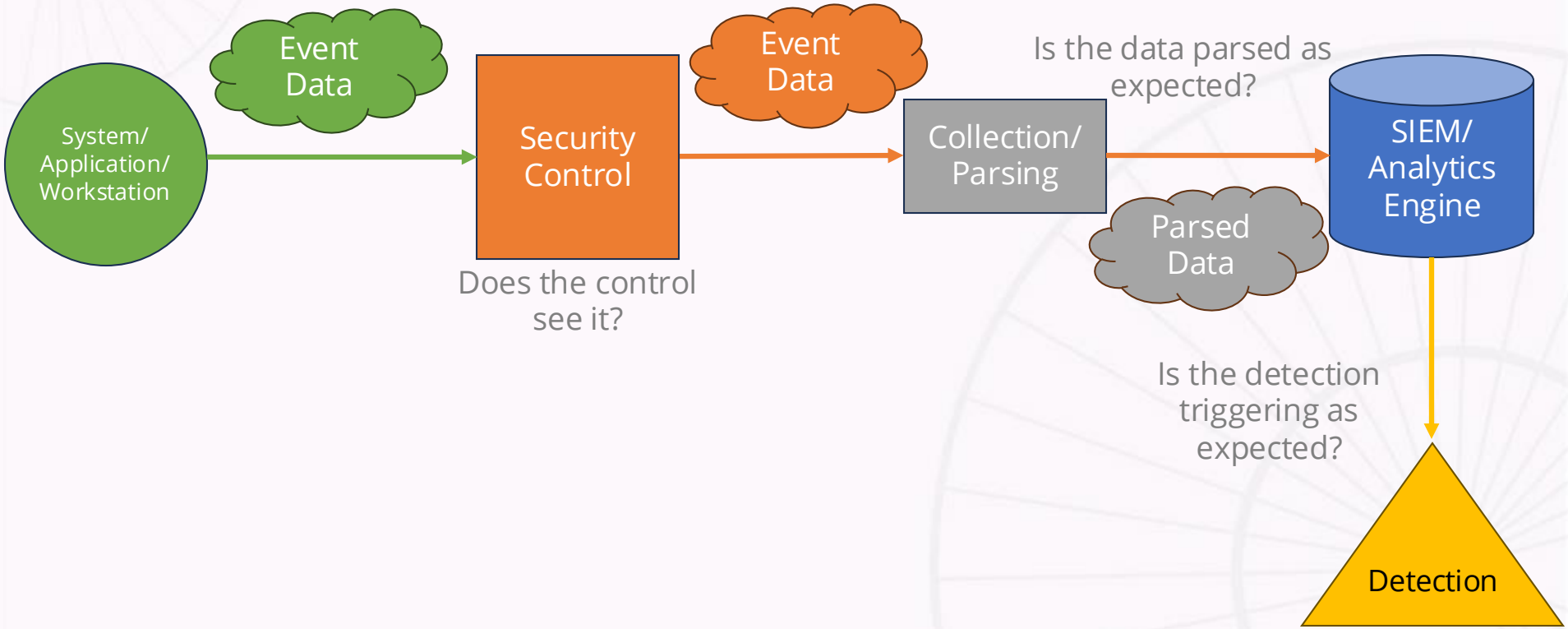
- Production v Development environments
- Platform coverage
- Staffing
- Maturity of current environment
- Scaling toward new platforms (like cloud)
- Understand your adversaries and threat posture

Share some ideas and methods to provide a path, not THE path forward

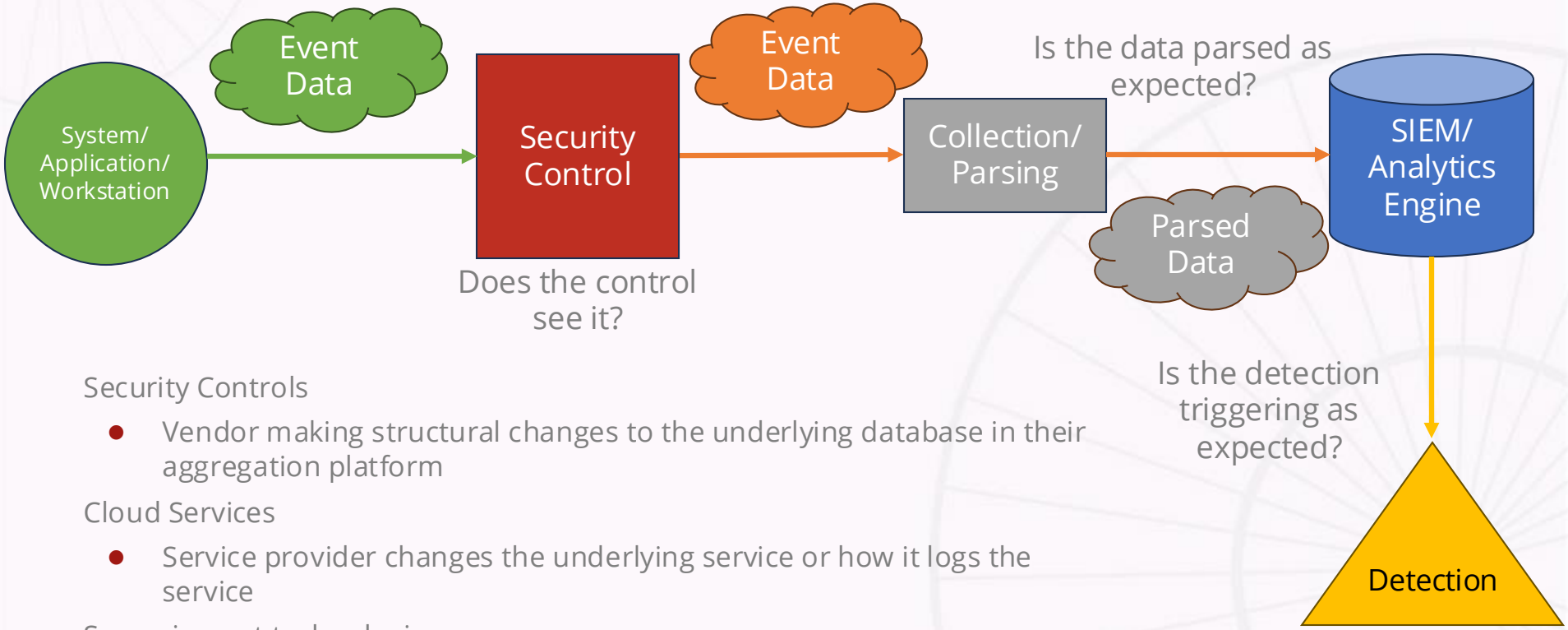
Our Goal

Build better detections while being able to refine them as environments evolve

Data Flow Through a SecOps Pipeline



Security Control Pain Points



Security Controls

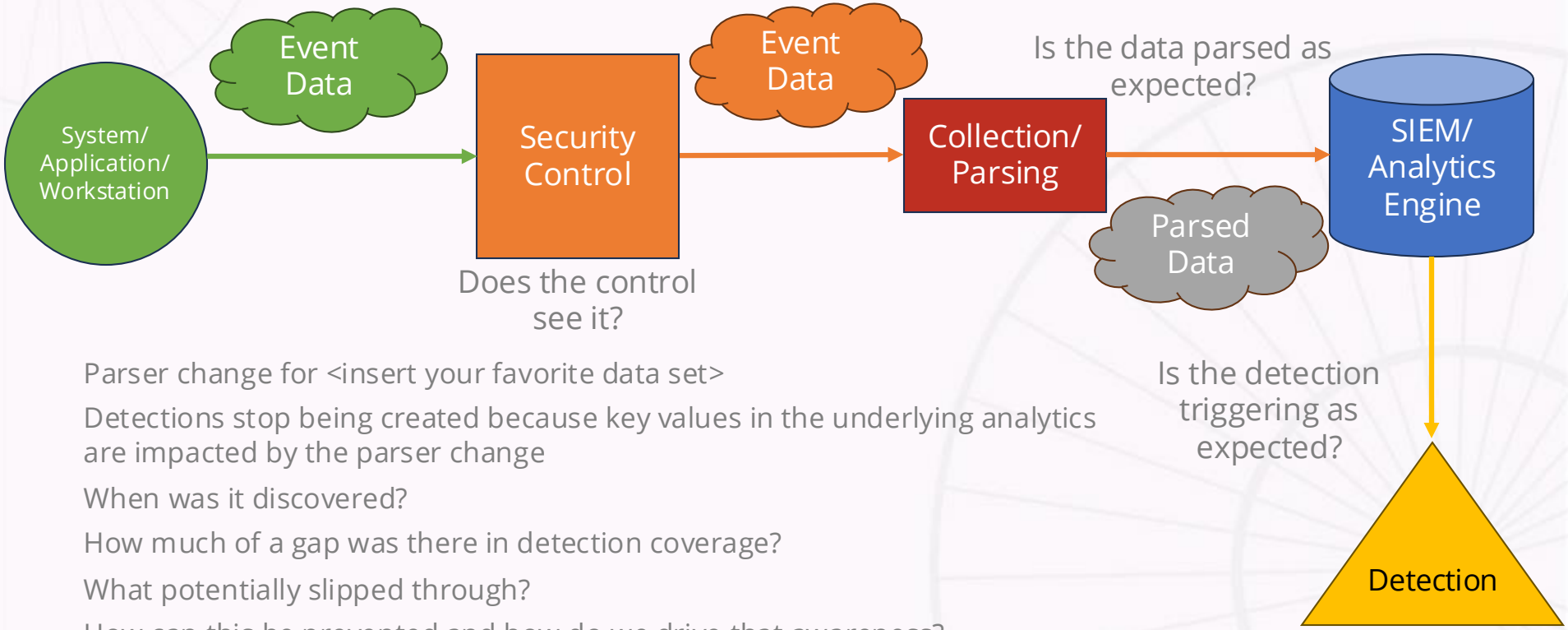
- Vendor making structural changes to the underlying database in their aggregation platform

Cloud Services

- Service provider changes the underlying service or how it logs the service

Swapping out technologies

Parsing Pain Points



Parser change for <insert your favorite data set>

Detections stop being created because key values in the underlying analytics are impacted by the parser change

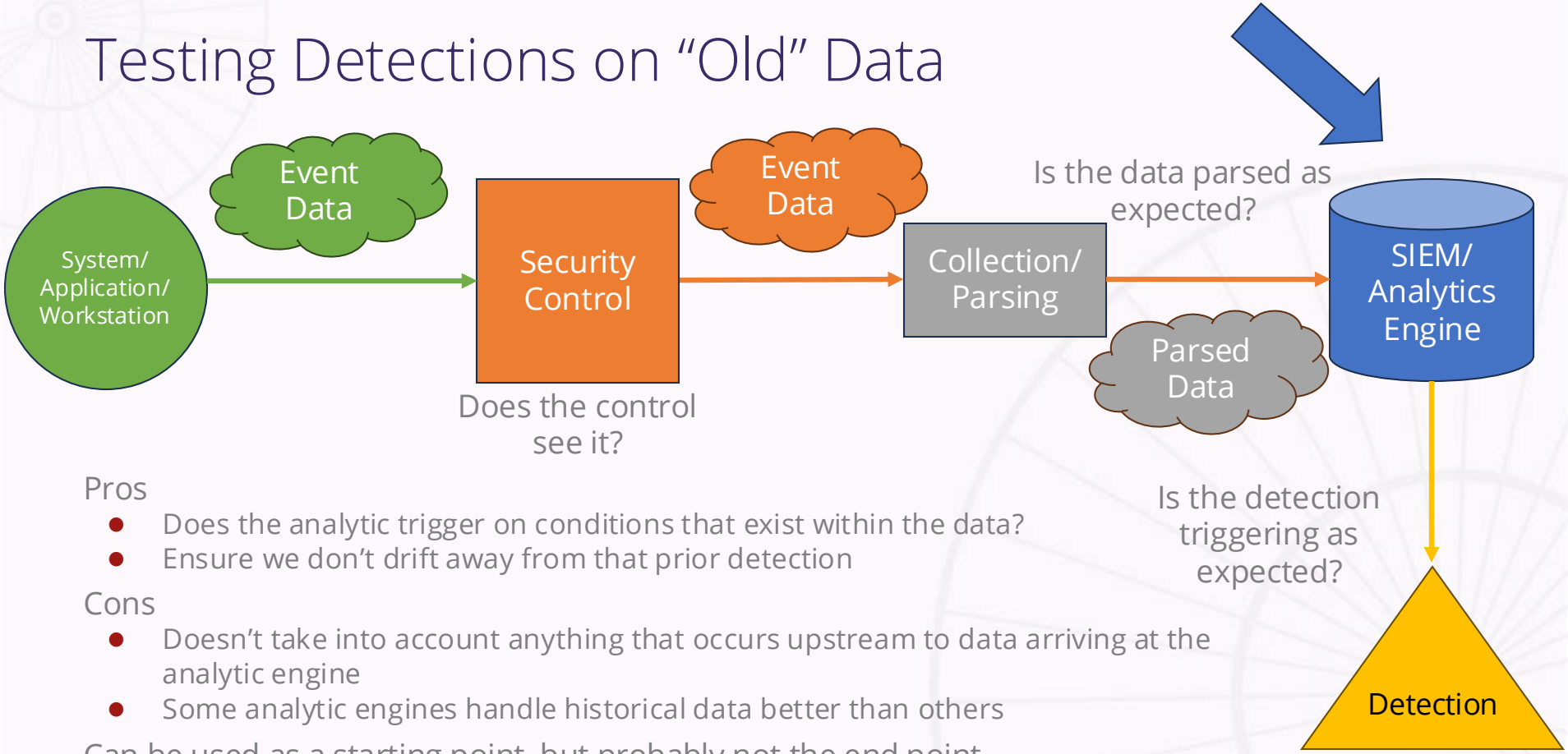
When was it discovered?

How much of a gap was there in detection coverage?

What potentially slipped through?

How can this be prevented and how do we drive that awareness?

Testing Detections on “Old” Data



Pros

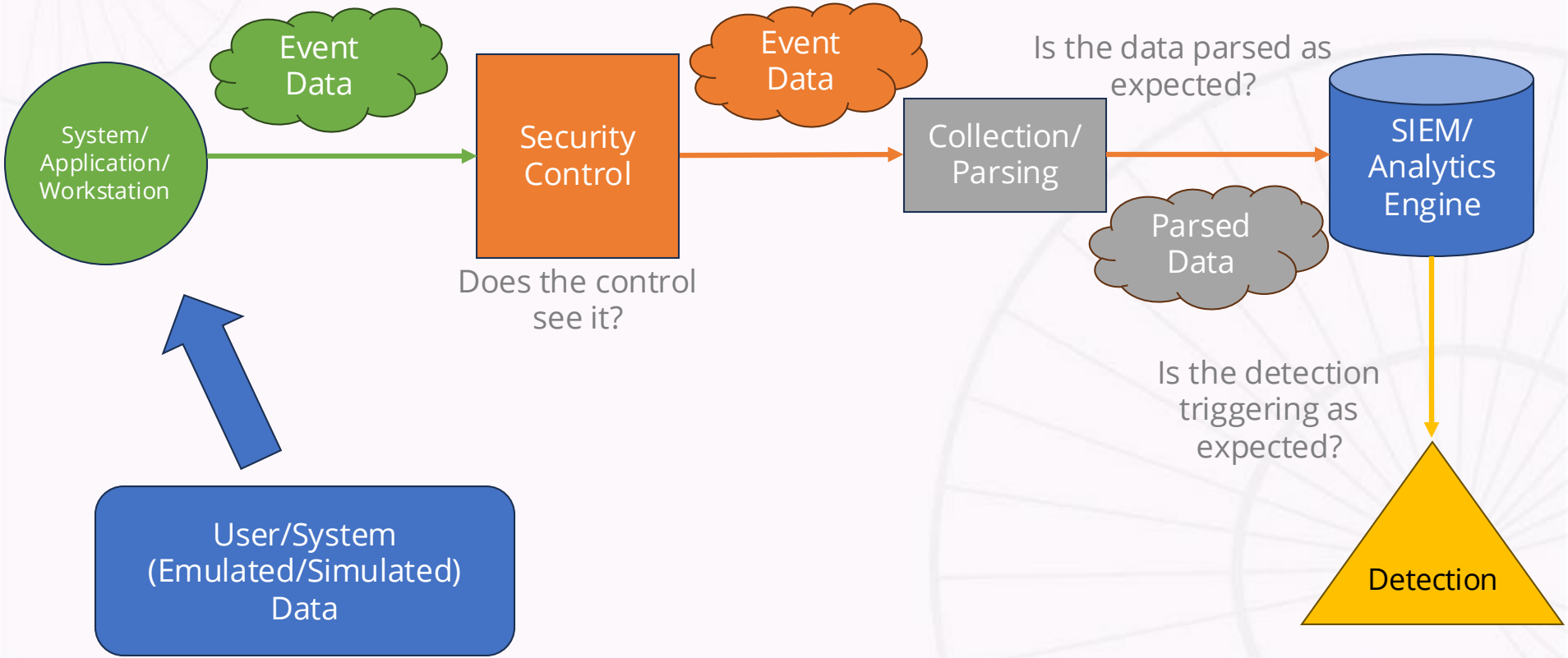
- Does the analytic trigger on conditions that exist within the data?
- Ensure we don’t drift away from that prior detection

Cons

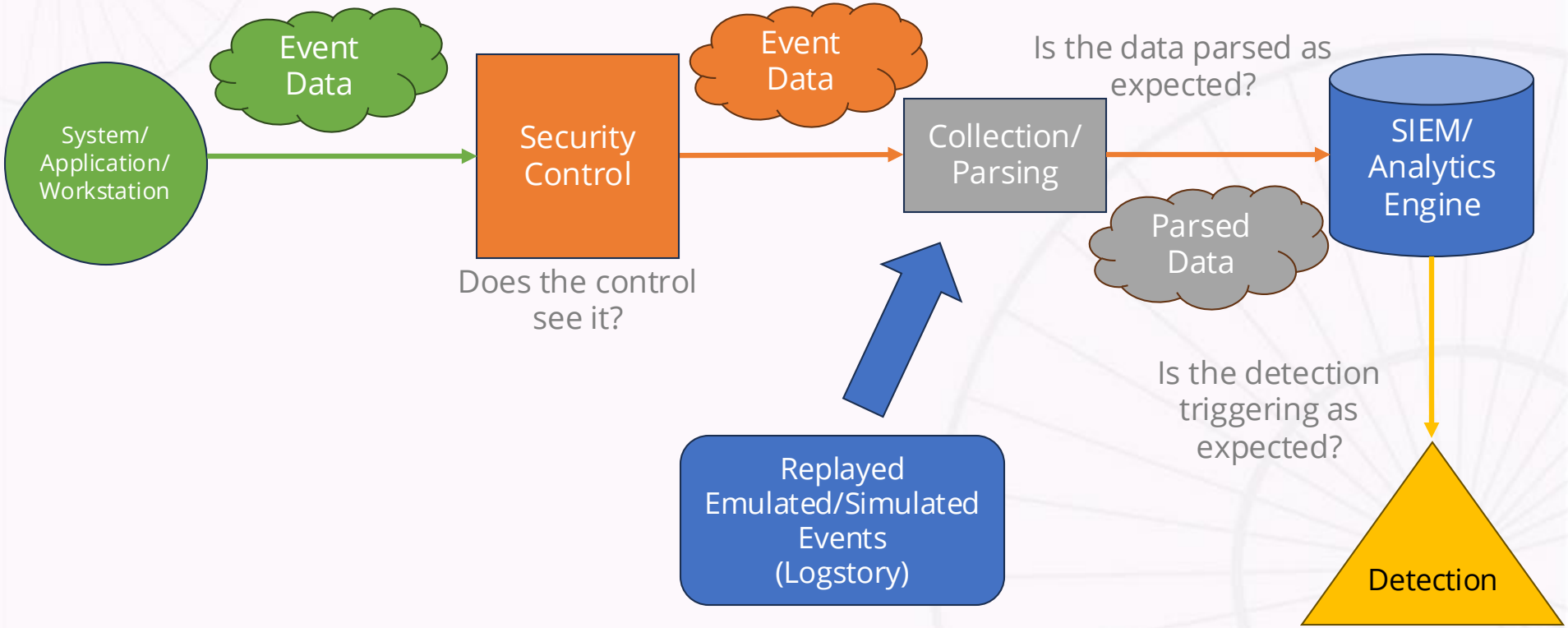
- Doesn’t take into account anything that occurs upstream to data arriving at the analytic engine
- Some analytic engines handle historical data better than others

Can be used as a starting point, but probably not the end point

Initial Run of an Emulation/Simulation



Emulation Replay



Continuous Testing with Emulation Replay

Can help address collection & parser issues

- Are fields/values missing or not aligned to analytic logic?
- Do our analytics continue to trigger?

Timestamps can be a challenge

- Late arriving data
- Some engines may ignore old data
- Existing rules may need to be re-pointed at the timestamps

Can be a great approach for learning - consistent environment with expected output

Does not address upstream issues like security controls changing

Continuous Testing with Sample Events

Could we just load sample sets of events that trigger analytics?

- Lighter data load rather than full data emulation
- As more detections are added, the overall system load will be negligible

If we aren't hunting or examining the full data set, why not just load the events that rules need to trigger?

Can be time consuming to go back through rules to create these sample sets

Extracting triggering events during rule development/build process isn't bad

Handling Advanced Analytics

Analytics with variable conditions to trigger introduce complexity in your sample set

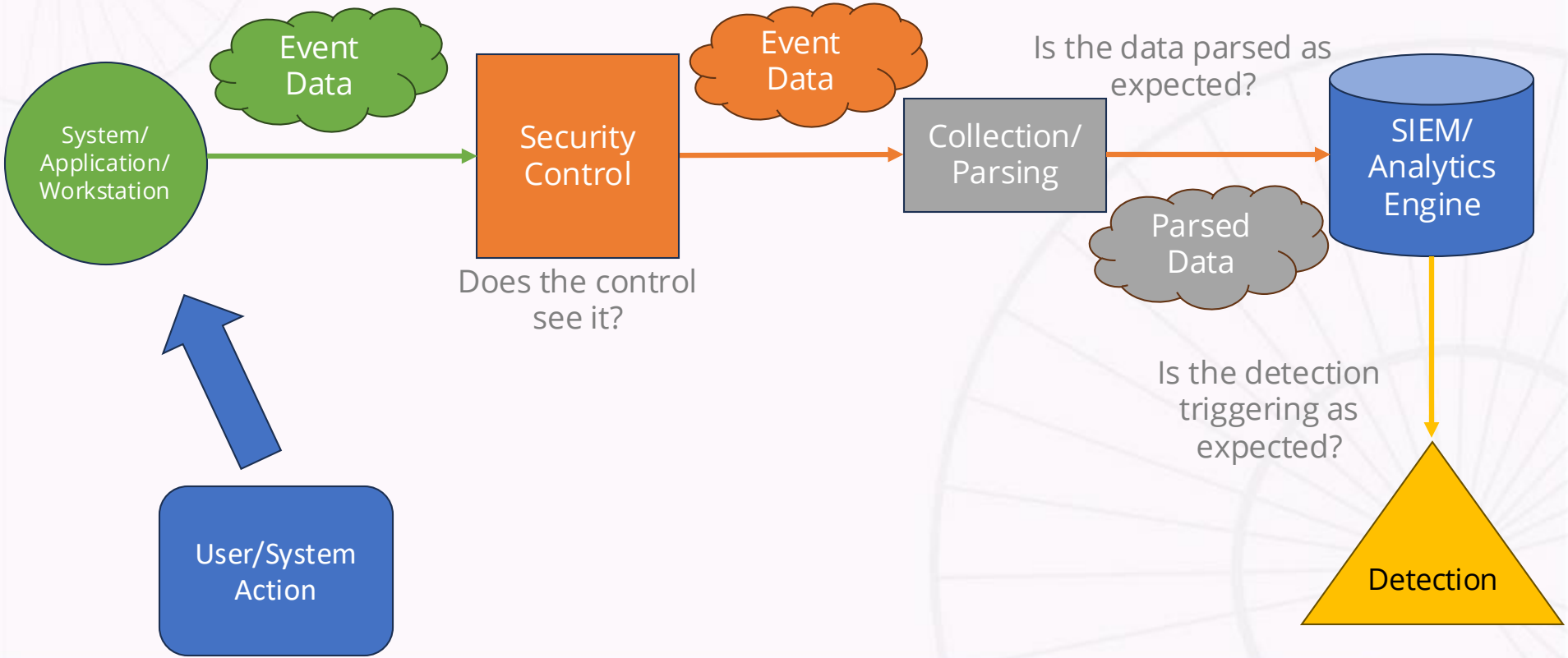
For instance, *Login and Subsequent Activity*

- Triggers on one specific event and a combination of four others
- Do we need to have examples of all the permutations to test against?
- Don't want to have to rely on just one of those and if something changes, then the rule goes silent

SIGMA rules have conditions like this as well

Existence of contextual data and watchlists can also impact tests

Continuous Testing by Executing Actions



Continuous Testing with Actions

Perform action at an endpoint that can flow through the pipeline

- If actions are generated but no data and/or detection is available, we can investigate along the pipeline
  - Architecture change
  - Security control change
  - Collection and parsing changes
  - Rule drift

Automation will be required to do this with any scale

Greater visibility across the pipeline also creates greater efforts to troubleshoot

Tooling

Atomic Red Team

- Aligned to MITRE ATT&CK
- One to many atomics per technique and subtechnique
- Supports Windows, Linux and MacOS as well as cloud

Invoke-AtomicRedTeam

- PowerShell wrapper to execute Atomic tests

```yaml
attack_technique: T1021.002
display_name: 'Remote Services: SMB/Windows Admin Shares'
atomic_tests:
- name: Map admin share
  auto_generated_guid: 3386975b-367a-4fbb-9d77-4dcf3639ffd3
  description: |
    Connecting To Remote Shares
  supported_platforms:
  - windows
  input_arguments:
    user_name:
      description: Username
      type: string
      default: DOMAIN\Administrator
    share_name:
      description: Examples C$, IPC$, Admin$
      type: string
      default: C$
    password:
      description: Password
      type: string
      default: P@ssw0rd1
    computer_name:
      description: Target Computer Name
      type: string
      default: Target
  executor:
    command: |
      cmd.exe /c "net use \\#{computer_name}\#{share_name} #{password} /u:#{user_name}"
    name: command_prompt
```

Detections

Where to start?

- Vendor provided detections
- Rules already built and deployed on my system
- Wanted to get a wider swath of rules so I tapped into SIGMA

Built a python script to search for tags to focus on specific techniques and types of rules

- Translated a SIGMA detections to YARA-L to get some good examples
- Leveraged Vertex AI to convert a larger detection set to YARA-L
- Reviewed results and tuned as needed

Testing Platform

Windows 2022 Server

- Sysmon config (same config as other systems of similar type)
  - Multiple configs may require multiple tests
- No additional EDR, but easily could have added one, just **detections**
- Designated specific user login for testing
  - Domain user with admin privileges to emulate an admin compromise

Set up logging to SIEM/Analytics Platform

- Added a PII filter for the user password

Test Preparation

Many atomics have prerequisites that need to be installed

`-checkreqs` and `-getprereqs` will automate much BUT NOT everything

Some prerequisites require manual action

- Installing Microsoft Office
- Software may no longer exist where the atomic expects it
- Atomic points to the zip file but requires you to run the installation
- Ruby version in the Atomic was not available, ended up installing 3.x manually

Tests can still fail even with prerequisites in place due to policies

- Default Atomic password doesn't have sufficient password complexity

Read The Atomic Test!!!

Fight the impulse to execute `invoke-atomic T1234-4` before reading the test

While testing T1021.001, my Remote Desktop Client would drop

- Port changes will cause this to happen

Understand the impact of the tests you are running

- IT troubleshooting and reviewing logs resolve many of these problems

Schedule cleanups to take place shortly after the test has run

- Cleanup commands will mitigate some of these problems
- Try to keep your system tidy

Customizing Tests

```
# Script expects that the pre-reqs are in place before adding here
$inputs = @{
    "ip_address" = "10.128.15.192"
    }

Write-Host "Test for Technique T1105" -ForegroundColor Yellow

# Non windows tests - 1-6
Sleep 10
Write-Host "Start Test - Time: $(date)" -ForegroundColor Red
#Run test 7 for technique
invoke-atomictest T1105-7 -InputArgs $inputs
```

Do we want to detect on the command line being issued or do we want to use other events?

You could modify the atomic with input values

- Mistakes modifying this will cause runtime errors

`-InputArgs` is a good way to customize atomic tests without having to change the atomic

- Passwords may be loaded into these inputs so be aware of that

Executing Actions - Getting Started

Identified some very noisy rules

- RDP rule generated 285 detections
  - Called for exception listing of permitted tools - parser issue
- Process access rule generated 1740 detections
  - Bad rule writing/conversion, OR instead of AND
- Remote login from Public IP
  - Required exceptions from systems RDP-ing into the network

Some rules didn't fire

- Password dumper rule
  - Latest parser had support for the 4656 event and added that process access mask value to rule

Trough of Disillusionment

Sharpening my Atomic Red Team efforts & converting SIGMA rules

Mapping Atomic Red Team Tests & Rules

- Some techniques align, many don't
- Identify overlaps in coverage - Can we tag the data to the rule/detection?
- Possible misses in coverage (no test to detect) - Need to build our own
- Need to get a feel for what detections have never fired
- Scripts run a hundred tests, how do we reconcile all the tests to all the rules

Process to Map Atomic Tests & Detections

Run a test, log is generated, rule is triggered - Rinse and repeat

Built a unit test powershell script with color coding and timestamps for tests to find underlying data for validation

Scripting the testing and adding start test times to trace the test to the logging and the associated detection(s)

Rolled these unit tests up

```
Done executing test: T1105-28 Nimgrab - Transfer Files
Cleanup
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

Executing cleanup for test: T1105-28 Nimgrab - Transfer Files
Done executing cleanup for test: T1105-28 Nimgrab - Transfer Files
Start Test - Time: 05/16/2025 13:58:02
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

Executing test: T1105-29 iwr or Invoke Web-Request download
Exit code: 0
Done executing test: T1105-29 iwr or Invoke Web-Request download
Cleanup
PathToAtomicsFolder = C:\AtomicRedTeam\atomics
```

Establishing a Testing Schedule

Service and Task Scheduler options are built into Invoke-AtomicRedTeam

- Task scheduler option would cause my system to reboot every 22 minutes or so
  - Traced it to a wmiprvse message, turned off the task
- Service was throwing PowerShell versioning errors

Created my own scripts

- Modular broken out by unit tests
- Logs tests to the execution log
- Includes cleanup commands
- Included times for tests to kick-off

```
# Non windows tests - 1-6
Sleep 10
Write-Host "Start Test - Time: $(date)" -ForegroundColor Red
#Run test 7 for technique
invoke-atomictest T1105-7 -InputArgs $inputs
```

Take Good Notes!

Tracking progress and tying tests to rules can be a bit daunting

Many rules had tactics and techniques, ones that didn't were updated

- What if the detection tactic/technique doesn't align to the Atomic test?

Added tags to rules to align Atomic tests to rules and associated detections

Developed a tracking spreadsheet of rules and tests

Atomic Test Tactic	Atomic Test Technique	Atomic Test Number	Atomic Test Name	Unit Test Created	Run Time Issues	Triggered		Rule Name
TA0002	T1059.001	1	T1059.001-1 Mimikatz	Yes		Yes	x	hacktool_mimikatz_execution
TA0002	T1059.001	2	T1059.001-2 Run BloodHound from local disk	Yes		No		
TA0002	T1059.001	3	T1059.001-3 Run Bloodhound from Memory using Download Cradle	Yes		No		
TA0002	T1059.001	4	T1059.001-4 Mimikatz - Cradlecraft PsSendKeys	Yes	Error	Yes		PowerShell Download from github.com/Power
TA0002	T1059.001	5	T1059.001-5 Invoke-AppPathBypass	Yes	The system cannot	Yes		Powershell Net Webclient Download

Making Atomics into Compounds

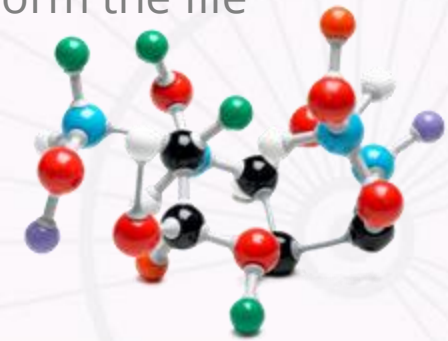
`Invoke-AtomicRunner -ListOfAtomics .\sequence.csv`

Provides a method to string a set of Atomic tests together in csv

- Could be a little lacking depending on how you want to string tests together
- *Copy remote file locally* and *curl download* had specific file locations defined that were not part of input arguments; would have to modify the Atomic
- Alternatively, build a PowerShell script to call the test, perform the file manipulation and then call the next test

Separate execution log by default for this

Cleanup command can be called after this is run - Use it!



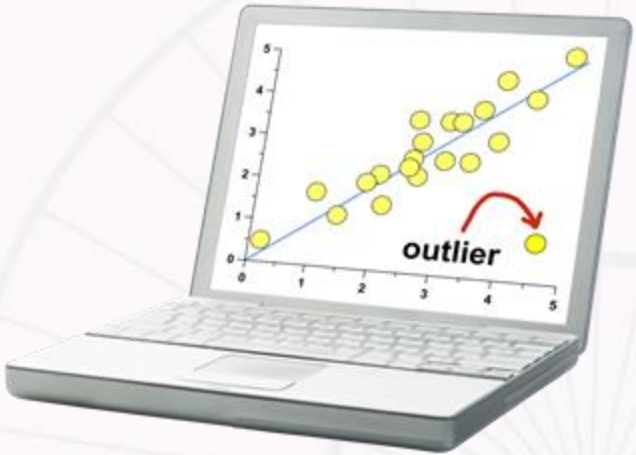
Exception Handling

Logging testing execution and monitoring for potential abuse

- Exception rule to detect other users executing tests from testing platform

Monitor for where the user accounts used for testing are seen beyond test systems and potentially time windows

- Accessing cloud services
- Crown jewel systems
- Tier 0 Infrastructure



Building New Rules for Tests

Be careful you don’t overfit detection logic

- Good for detecting the test, but not be great for detecting an adversary

Important to balance the precision to the intent or behavior

Any detection can fall victim to this

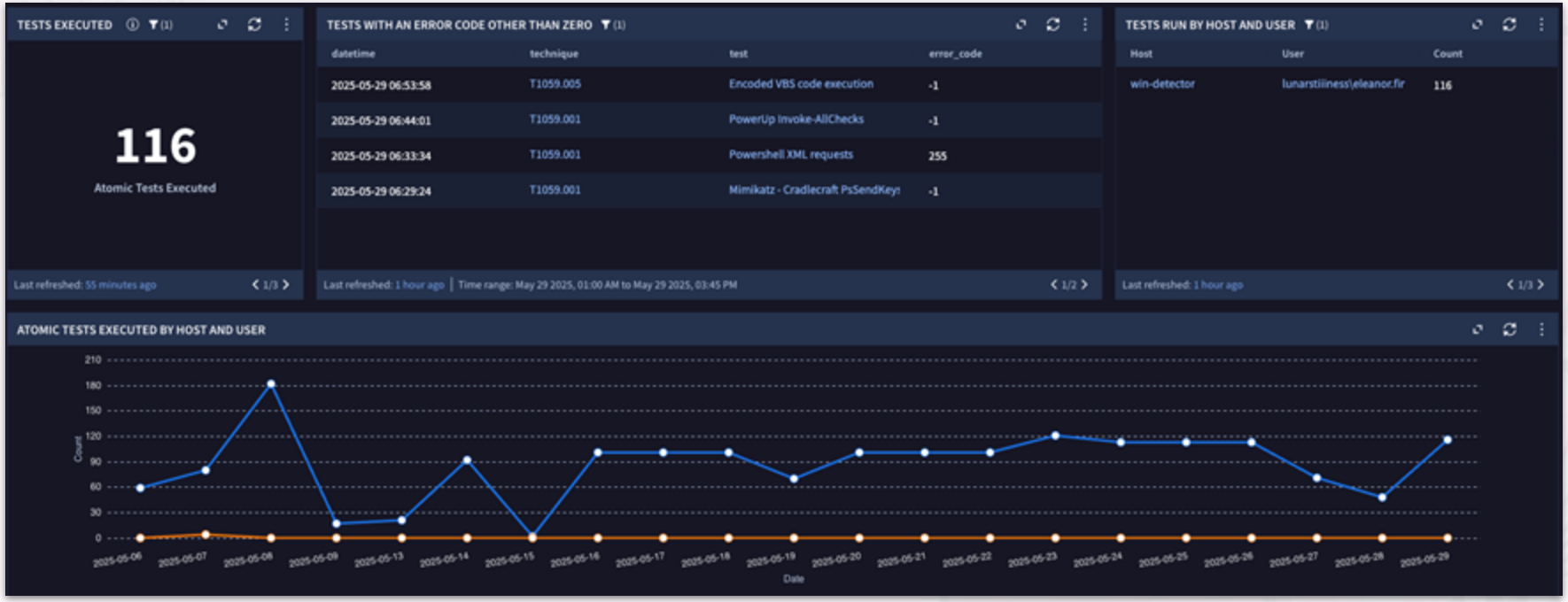
Don’t Create More Work for Your Analysts

Consider a way to mark Atomic tests with an internal notation to help suppress

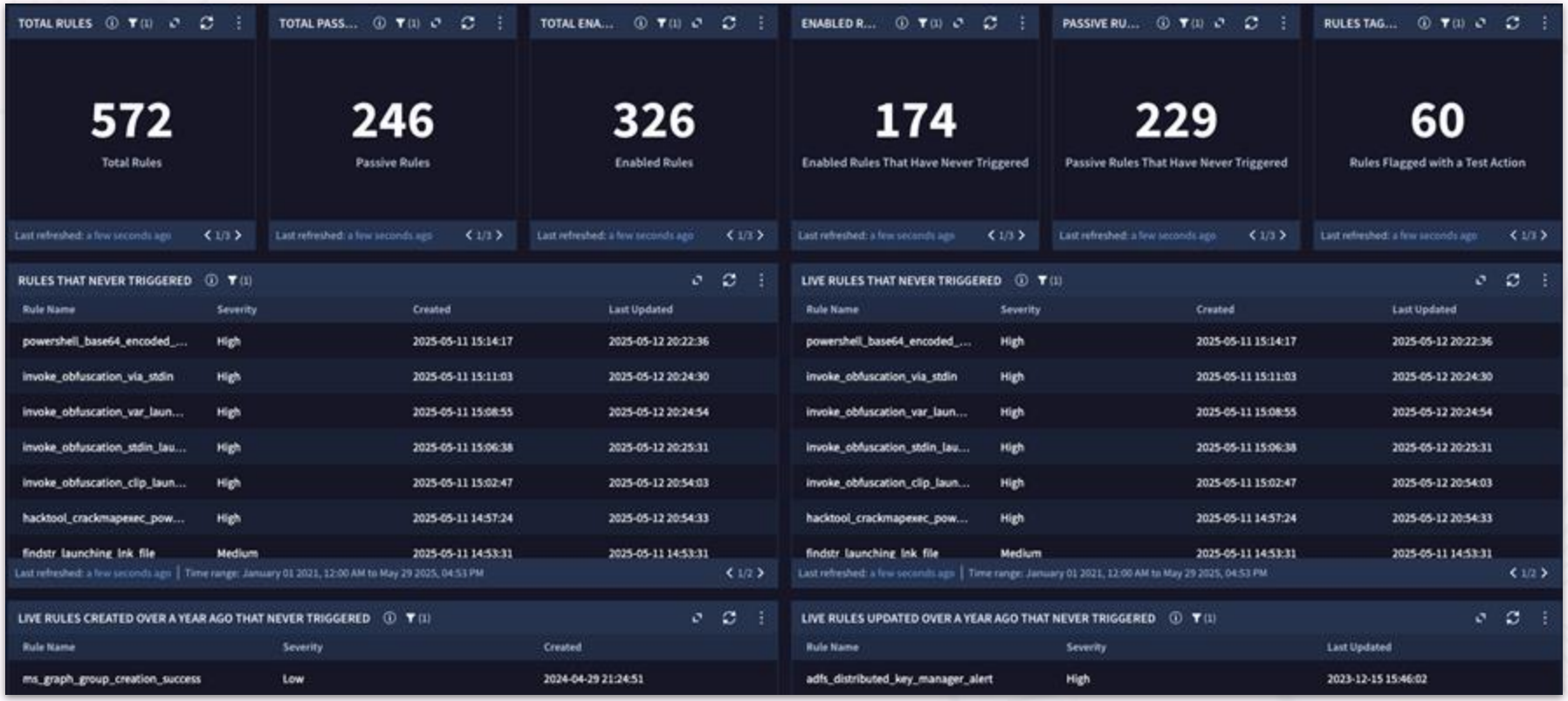
- Leveraged user and host to avoid incrementing risk scores
- Considered adding strings to Atomic tests and build those strings into rule logic to suppress
- Watchlists can also be used
- Consider playbooks for automatic case closing if needed as well

Extends to SOC Metrics as well

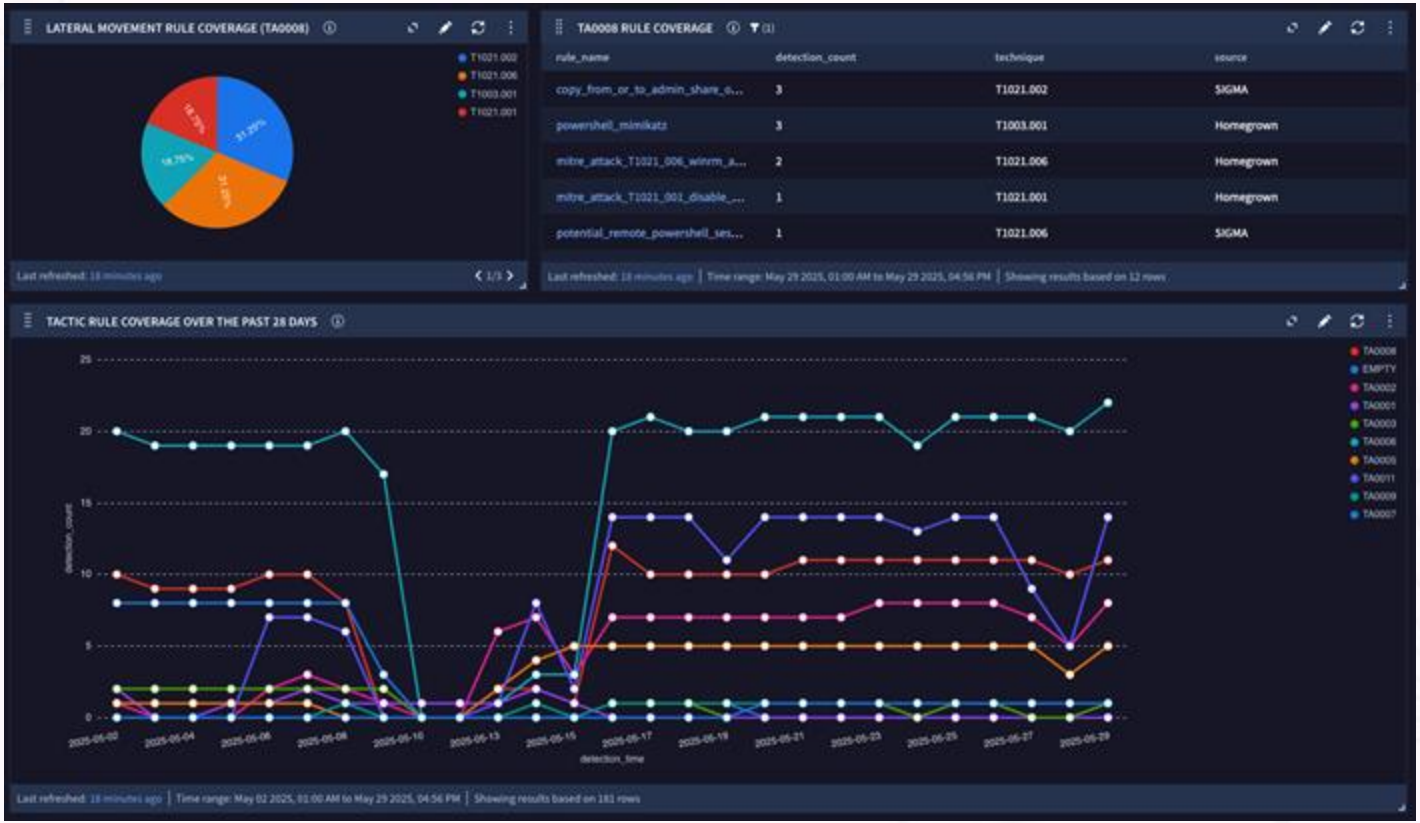
Measuring - Test Execution



Measuring - Rule Performance



Measurements - ATT&CK Coverage & Progress



Lessons Learned

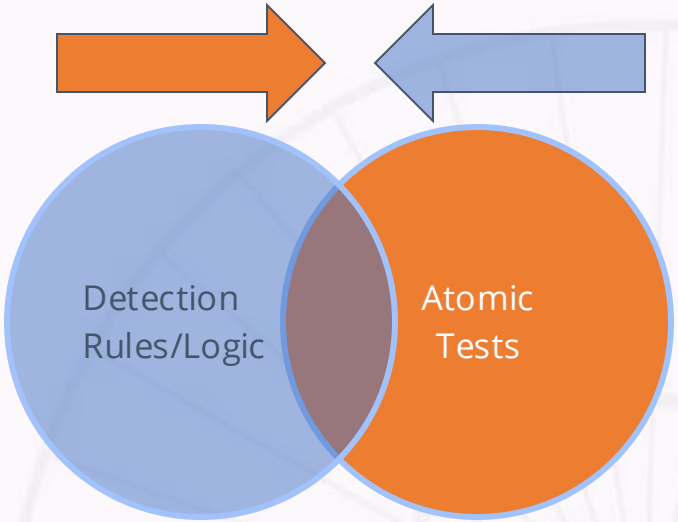
Many rules, many tests; but need to focus on bringing those circles closer to one another

Existing rule set contains multiple rules that trigger against the same atomic

- Probably need to retire some of those rules

Not all security controls are configured for the events that I want

- Needed to tune control configuration



Where Do I Start?

Prioritization is key, don't boiling the ocean

Where are your pain points?

MITRE Sightings Project

- Prioritize the most frequently seen techniques

Risk analysis

- T1105-32 File Download with SQLCMD.exe
  - Do I run MS-SQL?
  - Do I need to prioritize this?

Resources

Atomic Red Team: https://www.atomicredteam.io/atomic-red-team

Invoke-AtomicRedTeam: https://www.atomicredteam.io/invoke-atomicredteam

Logstory: https://github.com/chronicle/logstory

Sigma: https://github.com/SigmaHQ/sigma

Google Community Rules: https://github.com/chronicle/detection-rules

CTID Sightings Ecosystem: https://center-for-threat-informed-defense.github.io/sightings_ecosystem/

Thank You!

John Stoner
Google Cloud

https://www.linkedin.com/in/johnastoner/
@stonerpsu
@stonerpsu@infosec.exchange