

Modern detection engineering with Google SecOps

Google
Cloud
Next 25



Proprietary



David French

Staff Adoption Engineer
Google Cloud



John Stoner

Senior Security Consultant
Google Cloud

Agenda

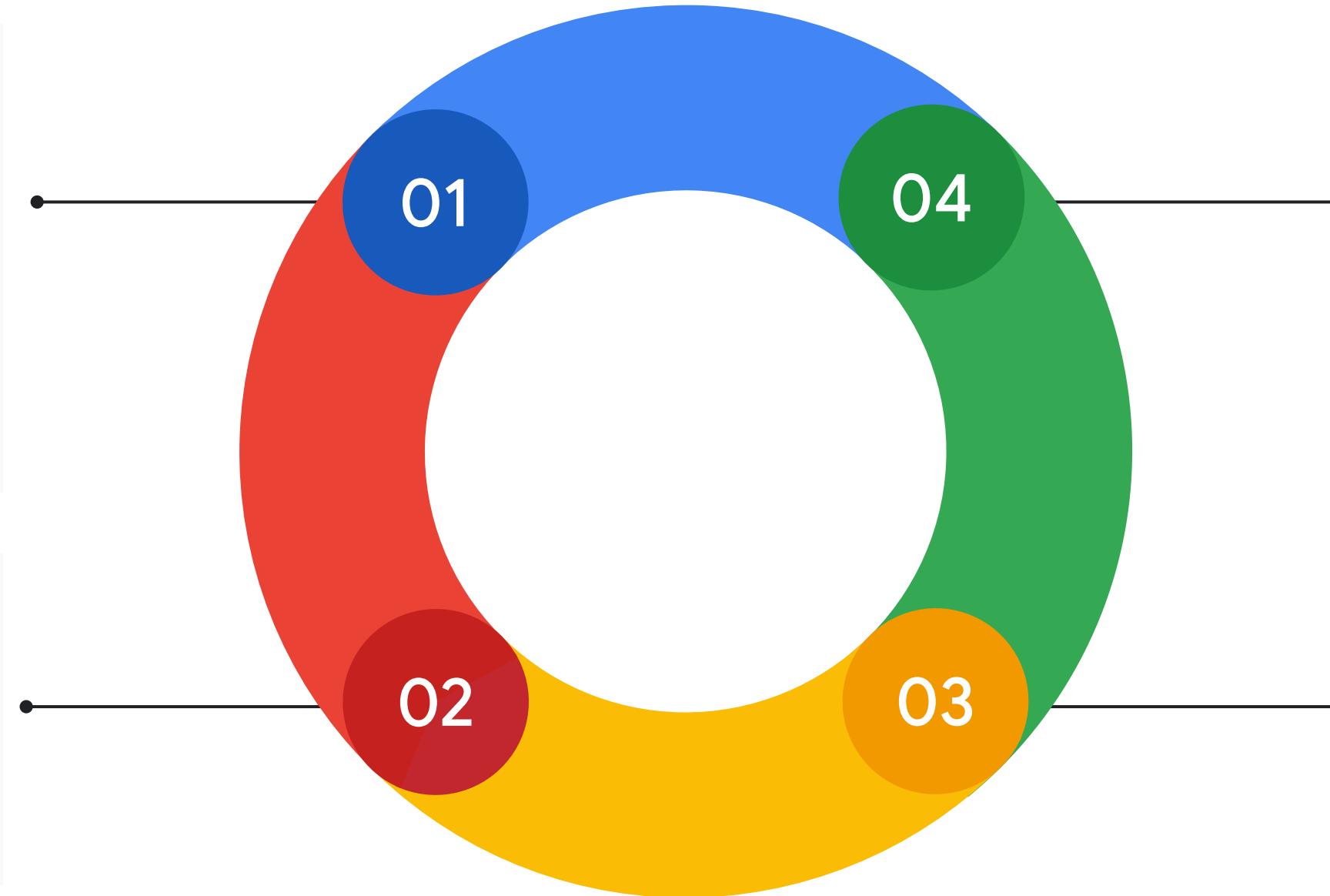


- 01 Detection Engineering
- 02 Curated Detections
- 03 Custom Rules
- 04 Detection-as-Code

01. Detection Engineering

Fundamental Concepts

Detection Engineering Workflow



Identify & Prioritize

Understand critical assets. Identify most relevant threats to your organization based on industry trends, past incidents, and potential impact. Assess detection coverage. Identify data sources.

Develop & Test

Develop detection logic to identify specific behaviors, patterns, or anomalies. Test & validate.

Monitor & Iterate

Continuously monitor detection effectiveness. Improve coverage and reduce noise. Incorporate feedback from security analysts, incident responders, and threat intel analysts.

Deploy & Tune

Implement detections in security tools and continuously refine them based on performance and observed activity.

Threat Intelligence Driven Detection Engineering



Informs Prioritization

Identify prevalent threats targeting our industry or technology stack

Understand attacker tactics, techniques, and procedures (TTPs) to focus detection efforts



IOCs are good but TTPs are better!

IOCs provide initial detection opportunities, they are often short-lived.

Focusing on detecting attacker TTPs leads to more resilient detections



Common Language

Leverage the MITRE ATT&CK framework to identify gaps in detection coverage against known adversary TTPs and adjust accordingly

Prioritization of Data Collection

-  Prioritize the onboarding of data sets based on threat intelligence
-  Configure preprocessing for log events (filter, redact, enrich, etc)
-  Log data that protects your most critical information should be prioritized - Crown Jewels
-  Detection development starts as soon as the first data is onboarded

02. Curated Detections

Curated Detections



Applied Threat Intelligence

Host and Network based IOCs from Active Security
Breaches & Investigations



Mandiant Frontline & Emerging Threats

Derived from Mandiant threat intelligence and findings in the field



Cloud Threats

Detections within GCP cloud infrastructure, Workspace and Chrome Enterprise Platform to protect your information



Ease of Implementation

Rule packages can be toggled to detection and/or alert and analyzed.



Risk Analytics

User and entity based analytics metrics to uncover outlier behavior



Customize with Exclusions

Provides the ability to tune rule packages

Applied Threat Intelligence

ALERTS

IOC MATCHES

IOC matches are generated by both Applied Threat Intelligence and any threat feeds your organization has provided.

View ⓘ

Prioritized IOCs



IOCs



Search...



01/03/2025 14:17:15

- 04/03/2025 14:17:15



IOC	TYPE	STATUS	GCTI PRI... ⓘ	CATEGORIES	ASSETS	ASSOCIATIONS	CAMPAI... ↑	LAST SEEN
f579524421f56badb2...	HASH_MD5	Match	⚠️ High	Campaign tracked by Mandi...	oscar, oscar.w...	NUMOZYLOD ↗ UNC4536 ↗	CAMP.23.052	2025-03-02T13:59:44.232
scarfponcho.com	DOMAIN	Match	⚠️ High	Observed within the ecosyst...	zenya-right-pc	LOKIBOT ↗	--	2025-03-02T09:31:45.000
17150a137c43235ad0...	HASH_MD5	Match	⚠️ High	Capable of encoding data us...	mikeross-pc	TONEDEAF ↗ UNC1907 ↗	--	2025-03-02T13:46:39.000
e323c6aee8b172b572...	HASH_MD5	Match	⚠️ High	Indicator was published in p...	mikeross-pc	CONTI ↗	--	2025-03-02T09:46:05.000
a55db6bfa7dedecf73...	HASH_MD5	Reviewed	❗ Active IR	Capable of killing a running ...	oscar, oscar.w...	SYSTEMBC.V2 ↗ UNC4393 ↗	--	2025-03-02T13:59:43.232

Applied Threat Intelligence

ALERTS IOC MATCHES

Welcome to Alerts and IoCs. Looking for alerts from other sources? Go to the [Legacy Enterprise Insights page](#)

Status != Closed Rule = ATI High Prior... (+1) Clear all Refresh Time: None (default) Showing: Last 3 days

STATE	NAME	RULE	PRIORITY	RISK SCORE	SEVERITY	CASE
Open	ioc:8e570e32acb99abfd0da...	ATI Active Breach Rule Match for File IoCs (SHA256) [Active Breach Priority Hos...	1 Critical	95 HIGH RISK	Critical	MimiKatz Command A...
Open	ioc:2fda6e766e1b5263d7d9...	ATI High Priority Rule Match for File IoCs (SHA256) [High Priority Host Indicators]	4 High	85 HIGH RISK	High	Msiexec.exe Installing a...
Open	ioc:a9ab5725d4e96e39f500...	ATI High Priority Rule Match for File IoCs (SHA256) [High Priority Host Indicators]	4 High	85 HIGH RISK	High	NTDS.dit Extraction via...
Open	ioc:227164b06f201b07a8b8...	ATI High Priority Rule Match for File IoCs (SHA256) [High Priority Host Indicators]	4 High	85 HIGH RISK	High	Msiexec.exe Installing a...

Cloud Threats



Rule Pack Configuration

 WINDOWS THREATS
Mandiant Intel Emerging Threats

Release Date: 2023-11-15 Log Sources: EDR Platform: WINDOWS Capacity: 125 Last Updated: 2025-02-20 (12 days ago) Author: Google Cloud Threat Intelligence

This rule set contains rules derived from Mandiant Intelligence Campaigns and Significant Events, which cover highly impactful geopolitical and threat activity, as assessed by Mandiant. This activity may include geopolitical conflict, exploitation, phishing, malvertising, ransomware, and supply chain compromises.

Settings

Rules	Status	Alerting
Precise ⓘ	<input checked="" type="button"/> Enabled	<input checked="" type="button"/> ON
Broad ⓘ	<input checked="" type="button"/> Enabled	<input checked="" type="button"/> ON

Resources

8 MOST ACTIVE RULES
Over last 30 days

- [NTDS.dit Extraction via Windows Volume Shadow Copy](#)
- [Rundll32 execute long filename](#).
- [Invoke-Expression \(IEX\) Obfuscation](#)
- [MimiKatz Command Arguments](#)
- [SEACACTUS Backdoor Command](#)

EXCLUSIONS ⓘ

Exclusion	Activity ⓘ	Status	Applied To	Manage
Emerging Threat 0	Enabled	Rule Set	View	

[View All](#)

Coverage ⓘ

MITRE Tactics

- TA0002 Execution
- TA0003 Persistence
- TA0004 Privilege Escalation
- TA0005 Defense Evasion
- TA0006 Credential Access
- TA0007 Discovery
- TA0008 Lateral Movement
- TA0009 Collection
- TA0010 Exfiltration

MITRE Techniques

- T0807 Command-Line Interface
- T0843 Program Download
- T1003 OS Credential Dumping
 - T1003.001 LSASS Memory
 - T1003.002 Security Account Manager
 - T1003.003 NTDS
- T1005 Data from Local System

Google Cloud Next

Proprietary

03. Custom Rules

Custom Rules



Build Your Own Rules

Rules editor with auto-complete, active syntax checker and integrated rule test capability



Entity Graph Integration

Leverage data within the entity graph to build detections that can use prevalence and GCTI data like TOR exit nodes and Remote Access Tools



Risk Analytics

User and entity based analytics metrics can be leveraged to build your own bespoke rules



Community Detection Content

Contains rules based on real world use cases as well as samples to assist you in building your own rules



Flexible Language

Over 60 functions available to assist you in working with your data



Retrohunt

Run a rule over existing events and generate detections for any matches

Developing Custom Rules in Google SecOps

Demo

1000

1000

11. [Google AdWords](#)
12. [Facebook](#)
13. [Twitter](#)
14. [LinkedIn](#)
15. [YouTube](#)
16. [Instagram](#)
17. [Pinterest](#)
18. [Tumblr](#)
19. [Snapchat](#)
20. [TikTok](#)

Приложение 1		
Составлено в соответствии с Правилами по оценке и классификации риска в сфере информационной безопасности Российской Федерации		
Наименование	ФИО	Должность
Горбунов Евгений Николаевич	Генеральный директор	
Адрес места нахождения	125009, г. Москва, ул. Садовая-Синявская, д. 10	
Номер телефона	+7 (495) 123-45-67	
Номер факса	+7 (495) 123-45-68	
Электронный адрес	evgeny.gorbunov@gorbunov.ru	
Номер документа	ДОК-001	Дата выдачи
Номер документа	ДОК-001	Дата выдачи

Using Functions in Your Rules Rules

Functions provide more flexibility when working with data

Great for handling date/time values, strings, type conversion
and much more!

```
$decoded_value =  
re.replace(strings.base64_decode(re.capture($process.target.process.command_line, `(?i)(?:-enc|-ec|-en)\s*(\S*)`)),`\0`, "")  
(  
    re.regex($decoded_value, `WebClient`) nocase or  
    re.regex($decoded_value, `WebRequest`) nocase  
)
```

Community-Driven Detection Content

1

Provides a starting point

Use these rules for threat hunting and for inspiration when building your own custom detections. Test, tune and customize them.

2

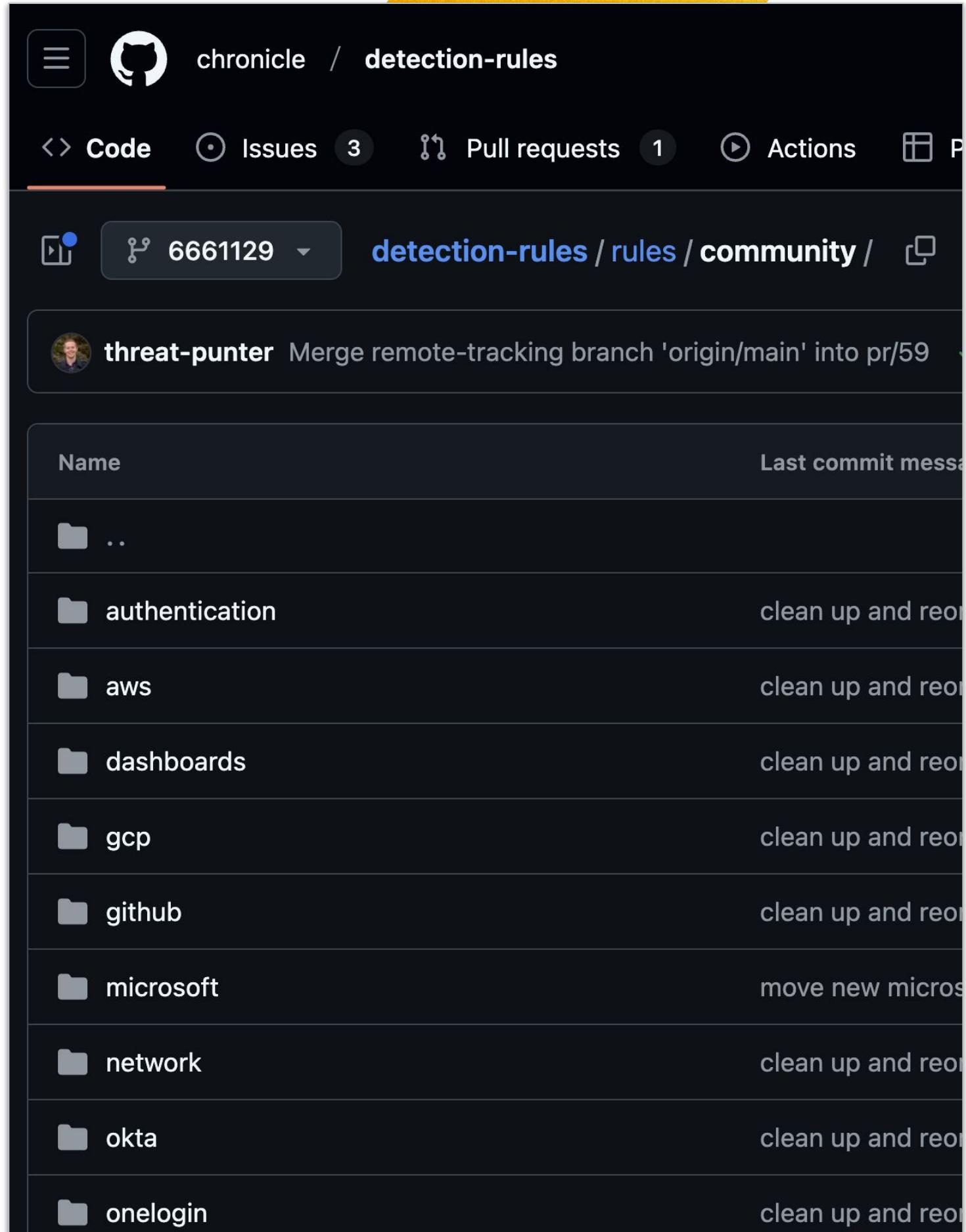
Demonstrates YARA-L usage

Many of these rules use the latest capabilities of YARA-L and are used for educational purposes.

3

We welcome contributions!

Contribution guidelines, a rule style guide, and example pull requests can be found in the GitHub repository.



Search to Rules Using Gemini AI

Demo



Google Security Operations

2025-03-04 20:06:57



Enter a hostname, domain, IP, URL, email, username, or file hash

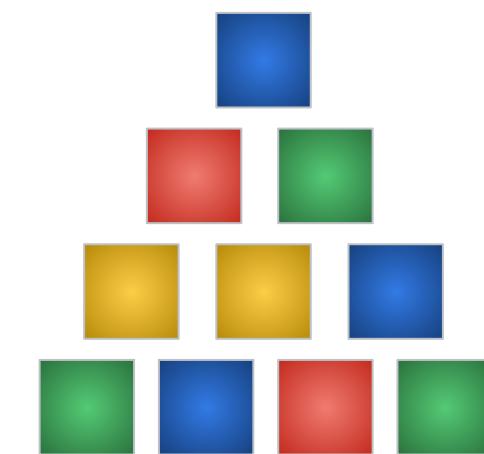
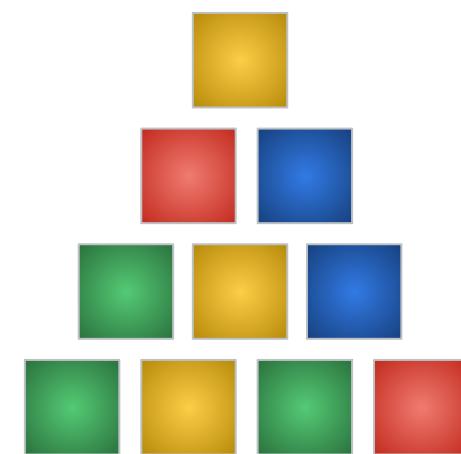
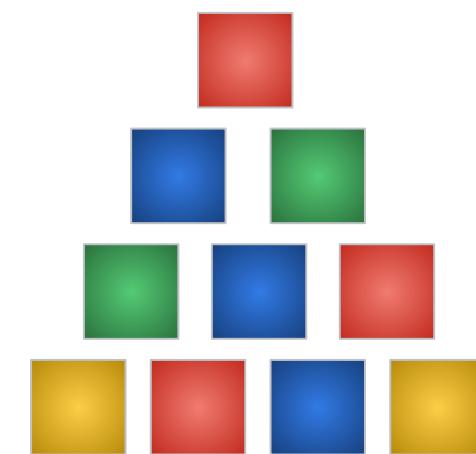
SEARCH

Total Log Entries: 3,886,638 Bytes Ingested: 6,043,853,280

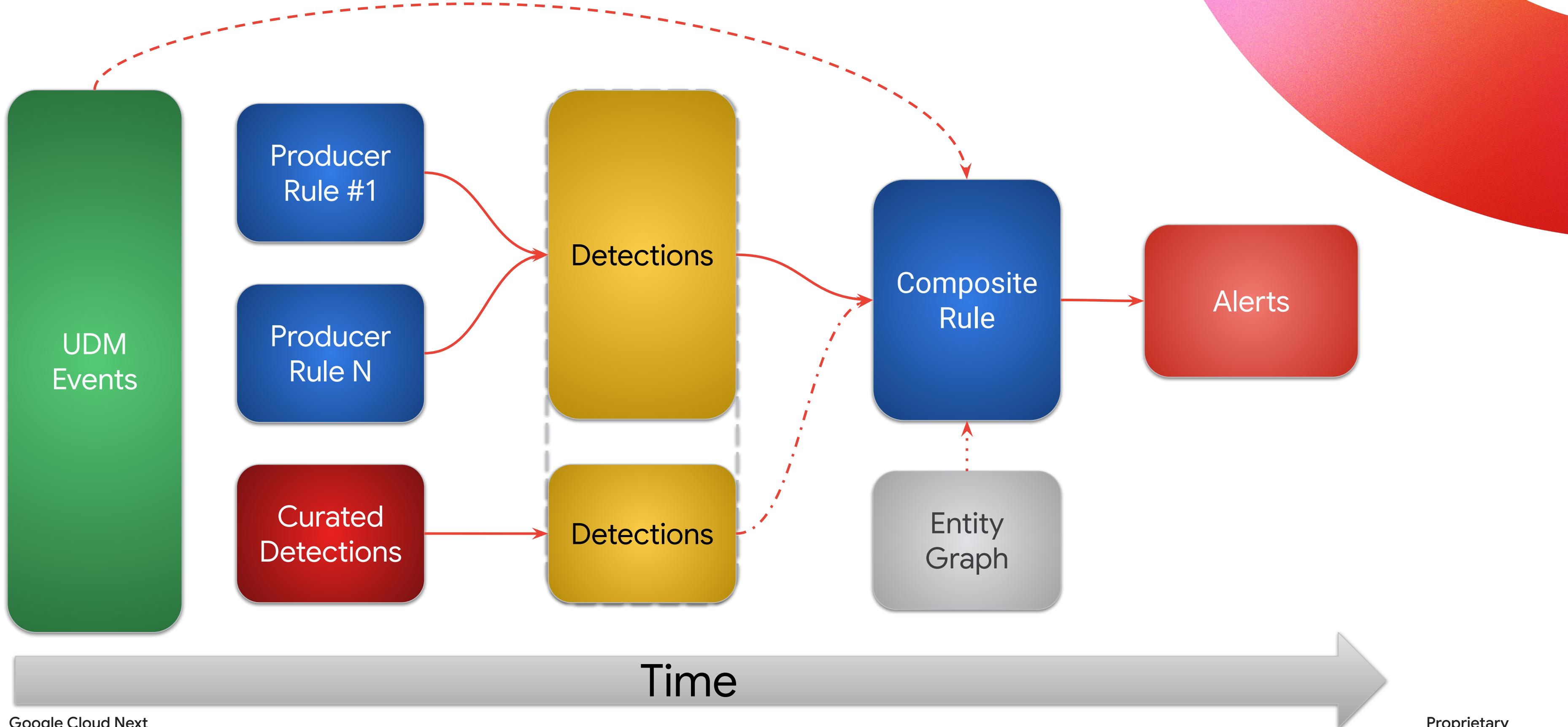
Composite Rules

Provides detection engineers and threat hunters the ability to build more flexible and advanced detections

- Break large, complex detections into smaller, atomic detections
- Re-use smaller detections over and over
- Leverage common values in the smaller detections to build composite detections



Composite Detections



Composite Detections Curated & Custom Rules

UDM Events

Detections from the curated detections package Mandiant Frontline Threats

Producer Rules

Custom rules that are Type - Single Event, Not Alerting with a label of producer

Reg Save HkLM
Sam Ss Dat

Reg Save HkLM
System Sy.Dat

Detections

Type - Single Event,
Not Alerting,
producer

Type - Single Event,
Not Alerting,
producer

Composite Rule
Common Match Variable of hostname and more than X
Frontline curated detections and Y custom detections

Alerts

Building a Composite Detection

```
events:  
    $detect_prod.detection.detection.rule_id = /^ru_/  
    $detect_prod.detection.detection.alert_state = "NOT_ALERTING"  
    $detect_prod.detection.detection.rule_type = "SINGLE_EVENT"  
    $detect_prod.detection.detection.rule_labels["type"] = "producer"  
    $detect_prod.detection.detection.detection_fields["hostname"] = $hostname  
  
    $detect_frontline.detection.detection.rule_set_display_name = "Mandiant Frontline Threats"  
    $detect_frontline.detection.detection.detection_fields["hostname"] = $hostname  
  
match:  
    $hostname over 4h  
  
outcome:  
    $risk_score = 60  
    $custom_rule_distinct_count = count_distinct($detect_prod.detection.detection.rule_id)  
    $frontline_rule_distinct_count = count_distinct($detect_frontline.detection.detection.rule_id)  
    $rules_triggered = arrays.concat(array_distinct($detect_prod.detection.detection.rule_name),  
        array_distinct($detect_frontline.detection.detection.rule_name))  
  
condition:  
    $detect_prod and $detect_frontline and $custom_rule_distinct_count > 3 and  
    $frontline_rule_distinct_count > 1
```

Composite Detections

Findings (1)

TIMESTAMP ↓	FINDING	NAME	RULE NAME	RISK SCORE ⓘ	CUS... ⓘ	FRONL... ⓘ	RULES_TRIGGERED ⓘ
2024-11-19T20:00:0...	TEST DETECTION (12)	hostname:win-adfs.lunarstiiness.c...	composite_custom_rules_with_ma...	60	8	2	producer_compress_data_...producer_impacket_wmie...producer_recon_credentia...+7 more

Findings (12)

TIMESTAMP ↑	FINDING	NAME	RULE NAME	RISK_SCORE ⓘ
2024-11-19T16:10...	DETECTION	netsh.exe launched by cmd.exe	producer_recon_environment_enumeration_netw...	0
2024-11-19T16:12...	DETECTION	systeminfo.exe launched by cmd.ex...	producer_recon_environment_enumeration_syste...	10
2024-11-19T16:12...	DETECTION	cmd.exe launched by powershell.ex...	producer_wmic_disk_discovery_T1082_cisa_report	15
2024-11-19T16:15...	DETECTION	cmd.exe launched by powershell.ex...	producer_recon_credential_theft_cisa_report	0
2024-11-19T16:15...	ALERT	reg.exe launched by cmd.exe	Reg Save Hkml Sam Ss Dat	20

Events (2)

TIMESTAMP	TYPE	SUMMARY	TARGET.PROCESS.COMMAND_LINE
2024-11-19T16:15:01.640	PROCESS_LAUNCH	reg.exe launched by cmd.exe	reg save hkml\sam ss.dat

producer_compress_data_lock_password_exfil_7zip
producer_impacket_wmiexec_cisa_report
producer_recon_credential_theft_cisa_report
producer_recon_environment_enumeration_active_directory
producer_recon_environment_enumeration_network
producer_recon_environment_enumeration_system
producer_recon_suspicious_commands_cisa_report
producer_wmic_disk_discovery_T1082_cisa_report
Reg Save Hkml Sam Ss Dat
Reg Save Hkml System Sy.Dat

Composite Detection - User Risk

```
events:  
    $detect_prod.detection.detection.outcomes["principal_user_userid"] = $user  
match:  
    $user over 24h  
outcome:  
    $risk_score = 60  
    $uniq_detection_count = count_distinct($detect_prod.detection.detection.rule_id)  
    $total_detection_count = count($detect_prod.detection.detection.rule_id)  
    $rules_triggered = array_distinct($detect_prod.detection.detection.rule_name)  
    // sum of the risk score to measure against the threshold  
    $cumulative_risk_score = sum($detect_prod.detection.detection.risk_score)  
condition:  
    $detect_prod and $cumulative_risk_score >= 90
```

Composite Detection - User Risk

TIMESTAMP	FINDING	NAME	RULE NAME	CUMUL...	UNIQ_DETE...	RULES_TRIGGERED
▼ 2024-11-20T00:00:00.000	TEST DETECTION (10)	user:tim.smith_admin	composite_cumulative_risk_scor...	175	7	producer_compress_data_lock_password... producer_recon_credential_theft_cisa_re... producer_recon_environment_enumerati...
Findings (10)						
	TIMESTAMP ↓	FINDING	NAME	RULE NAME	DETECTION.RISK_SCORE	
➤ 2024-11-19T16:30:00.000	DETECTION	cmd.exe launched by powershell.exe	producer_recon_credential_theft_cisa_report	10		
➤ 2024-11-19T16:28:30.000	DETECTION	cmd.exe launched by powershell.exe	producer_recon_environmentEnumeration_active_directory	15		
▼ 2024-11-19T16:27:00.000	DETECTION	cmd.exe launched by powershell.exe	producer_recon_environmentEnumeration_system_cisa	10		
Events (10)						
	TIMESTAMP	TYPE	SUMMARY	PRINCIPAL.USER.USERID	TARGET.PROCESS.COMMAND_LINE	
2024-11-19T16:12:05.116	PROCESS_LAUNCH	cmd.exe launched by powershell.exe	tim.smith_admin	"cmd.exe" /c wmic product list brief		
2024-11-19T16:12:13.933	PROCESS_LAUNCH	cmd.exe launched by powershell.exe	tim.smith_admin	"cmd.exe" /c wmic baseboard list full		
2024-11-19T16:12:48.933	PROCESS_LAUNCH	cmd.exe launched by powershell.exe	tim.smith_admin	"cmd.exe" /c wevtutil qe security /rd:true /f:text /q:		
2024-11-19T16:14:09.859	PROCESS_LAUNCH	cmd.exe launched by powershell.exe	tim.smith_admin	"cmd.exe" /c reg query hklm\software\realvnc		

Visualizing Detections



04. Detection-as-Code

Detection-as-Code is a methodology that applies software engineering principles to the creation and management of detection content

Motivations for Adopting DaC



Collaboration

A group of practitioners with unique insights working together results in more effective rules



Change Management

Every modification is tracked, reviewed, and approved leaving behind a clear audit trail



Automation

CI/CD pipeline automates the testing and deployment of detection rules



Version Control

Changes to detection content are tracked in git, which makes it easy to roll back changes if needed



Scalability

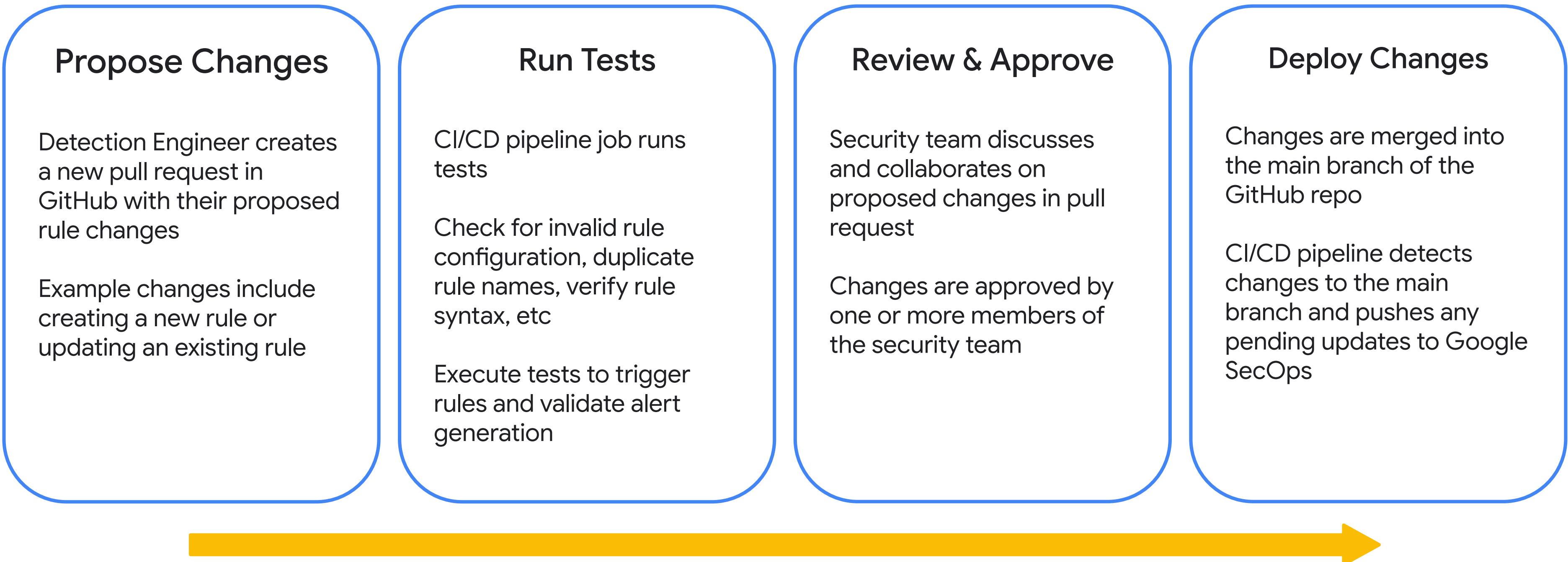
Managing a library of rules across multiple tools, environments, and customers is easier



Knowledge Management

Comprehensive documentation ensures that the detection's strategy, triage & response steps, assumptions, blind spots, etc are available

Typical DaC Workflow



Creating a New Rule (1)

The screenshot shows a GitHub Pull Request interface with the following details:

- Pull requests** tab is active, showing 3 pull requests.
- Title:** add new github rule #26
- Author:** threat-punter
- Destinations:** merge 1 commit into `main` from `new-rule/add-new-github-rule`
- Files changed:** 2 files
- Conversation:** 0
- Commits:** 1
- Checks:** 1
- Changes from all commits:** 3 changes in `rule_config.yaml`
- File filter:** Conversations
- Filter changed files:** rule_config.yaml
- rule_config.yaml:** 243 lines added, 238 lines removed.

```
@@ -238,3 +238,6 @@ test_rule_1:  
 238 238 revision_id: v_1730221819_830255000  
 239 239 run_frequency: LIVE  
 240 240 type: SINGLE_EVENT  
 241 + github_sso_configuration_modified:  
 242 + enabled: true  
 243 + alerting: true
```

- rules/github_sso_configuration_modified.yaral:** 52 lines added, 0 lines removed.

```
@@ -0,0 +1,52 @@  
 1 + rule github_sso_configuration_modified {  
 2 +  
 3 +   meta:  
 4 +     author = "Google Cloud Security"
```

Creating a New Rule (2)

The image consists of three vertically stacked screenshots of a GitHub pull request interface, each with a blue circle containing a number indicating a step in the process:

- Screenshot 1:** Shows the pull request status. It has a red 'Review required' badge with an 'X'. Below it, a green 'All checks have passed' badge with a checkmark and '1 successful check'. A 'Run Tests / run-unit-tests (push)' check is listed as successful. There are 'Required' and 'Details' buttons.
- Screenshot 2:** Shows the review history. It displays a comment from 'dandye' saying 'LGTM' with a smiley face emoji. There is also a link to 'View reviewed changes'.
- Screenshot 3:** Shows the merge button at the bottom of the pull request page. The button is green and says 'Merge pull request'. Below it, a note says 'You can also [open this in GitHub Desktop](#) or view [command line instructions](#)'.

Creating a New Rule (3)

✓ Update rules in Google SecOps based on files in main branch

```
1 ►Run python -m rule_cli --update-remote-rules
23
23 11-Nov-24 19:11:21 UTC | INFO | <module> | Rule CLI started
24 11-Nov-24 19:11:21 UTC | INFO | <module> | Attempting to update rules in Google SecOps based on local rule files
25 11-Nov-24 19:11:21 UTC | INFO | update_remote_rules | Attempting to update rules in Google SecOps based on local rule files
26 11-Nov-24 19:11:21 UTC | INFO | update_remote_rules | Loading local files from /home/runner/work/detection-engineering-demo-1/detection-engineering-
27 11-Nov-24 19:11:21 UTC | INFO | load_rule_config | Loading rule config file from /home/runner/work/detection-engineering-demo-1/detection-engineering-
   /rule_config.yaml
28 11-Nov-24 19:11:21 UTC | INFO | load_rules | Loaded 21 rules from /home/runner/work/detection-engineering-demo-1/detection-engineering-demo-1/rules
29 11-Nov-24 19:11:21 UTC | INFO | update_remote_rules | Attempting to retrieve latest version of all rules from Google SecOps
```

```
40 11-Nov-24 19:11:27 UTC | INFO | update_remote_rules | Logging summary of rule changes...
41 11-Nov-24 19:11:27 UTC | INFO | update_remote_rules | Rules created: 1
42 11-Nov-24 19:11:27 UTC | INFO | update_remote_rules | created github_sso_configuration_modified (ru_5e5299fa-2312-4298-8153-7a75ef7e3ce7)
43 11-Nov-24 19:11:27 UTC | INFO | update_remote_rules | Rules new_version_created: 0
44 11-Nov-24 19:11:27 UTC | INFO | update_remote_rules | Rules enabled: 1
45 11-Nov-24 19:11:27 UTC | INFO | update_remote_rules | enabled github_sso_configuration_modified (ru_5e5299fa-2312-4298-8153-7a75ef7e3ce7)
46 11-Nov-24 19:11:27 UTC | INFO | update_remote_rules | Rules disabled: 0
47 11-Nov-24 19:11:27 UTC | INFO | update_remote_rules | Rules alerting_enabled: 1
48 11-Nov-24 19:11:27 UTC | INFO | update_remote_rules | alerting_enabled github_sso_configuration_modified (ru_5e5299fa-2312-4298-8153-7a75ef7e3ce7)
```

Creating a New Rule (4)

The screenshot shows the Google SecOps Rules & Detections interface. The top navigation bar includes the Google SecOps logo and the current page title, "Rules & Detections". Below the navigation is a horizontal menu with four tabs: "RULES DASHBOARD", "RULES EDITOR" (which is currently selected), "CURATED DETECTIONS", and "EXCLUSIONS". On the left side, there is a sidebar with various icons and a "NEW" button. The main content area displays a list of rules, with the first rule, "github_sso_configuration_modified", highlighted by a red box. To the right of the list is a code editor window showing the JSON configuration for this rule. The configuration includes fields like status, creation date, and a warning message about saving edits while the rule is running. The JSON code is as follows:

```
1 rule github_sso_configuration_modified
2
3 meta:
4   author = "Google Cloud Security"
5   description = "Detects when changes are made to GitHub SSO configuration"
6   rule_id = "mr_65a0f413-7f77-4039-8e0d-000000000000"
7   rule_name = "GitHub SSO Configuration Modified"
8   assumption = "Your GitHub enterprise has GitHub SSO enabled"
9   type = "alert"
10  severity = "High"
11  priority = "High"
```

Implementation Considerations

Source of truth for detection content

For example, GitHub or Google SecOps

Should your pipeline overwrite any changes made outside your DAC pipeline?

Change control

If all changes should go through your DAC pipeline, should the permission to modify files in the UI be restricted?

Monitoring for rule changes

Is the behavior of your content being modified outside of your DAC pipeline suspicious? You can detect this behavior using Google SecOps

Resources



SecOps Community



Umbrella site with loads of resources to engage with others on all things SecOps!



New to Google SecOps Blog

Blog series with over 50 blogs on building content using many of the tools we discussed today!



How To Videos

Over 40 videos that demonstrate how to build a rule, use functions, apply metrics to rules, integrate entity graph data and more!



Community-Driven Detection Content

Repository of YARA-L rules that can be used to accelerate adoption as well as share & collaborate on detection content with community



Forums

Not clear on how to write that last line of YARA-L?
Not sure what the best function to use is? Come and engage the community and ask!



Community Blog

Everything from threat research to integrating with GCP audit logs, Detection-as-Code how tos, and technical tutorials and more!

Final Thoughts...

-  Leverage threat intelligence to focus detection engineering
-  Curated detections provide rules to get you started - Cloud, Windows, UEBA, Active Breaches and more!
-  Generative AI can help get you started with detection logic
-  Functions and Entity Graph provide additional flexibility to craft your own rules with more on the way!
-  It's easy to implement Detection-as-Code with Google SecOps

Your feedback is greatly appreciated!



**Complete the session
survey in the mobile app**

Thank you