



Monthly Partner Series

# Intro to Advanced Rule Management

May 2023

# Agenda

- Chronicle SIEM Release Notes
- Elements of Advanced Rules
- Managing those Elements through APIs
- Example layout of Rules git repo
- Leveraging Rule versioning to Recover

# Moving FAST!

We have limited time, so many topics will be touched but not covered in detail.

- Check out the Rules labs on the Partner Portal.
- Ask your PE about the Rules Workshop and the Entities & IOCs workshop
- Attend one of our Advanced Enablement Bootcamps.
- Read: <https://medium.com/@thatsiemguy>  
<https://chronicle.security/blog/?filters=new-to-chronicle-series>
- Special thanks to Eugene Dimarsky, John Stoner, Serhat Gulbetekin, and Oreste Dimaggio

# Chronicle Release Notes

## Deprecated

Chronicle forwarder executable for Windows is officially deprecated. The code has not been officially updated since 2022, except for specific break fixes. Use another collection mechanism for EVTX export. The docker container can receive on TCP/UDP/PCAP as well as local files (not the Event Viewer).

<https://cloud.google.com/chronicle/docs/install/docker-forwarder-windows>

## Feature

You can now install and configure [Chronicle forwarder for Windows on Docker](https://cloud.google.com/chronicle/docs/install/docker-forwarder-windows). This Docker installation provides better security through isolation and the container distribution mechanism can be private and separate for Google Cloud and customers. This release also includes the following updates:

<https://cloud.google.com/chronicle/docs/install/docker-forwarder-windows>

- The forwarder signing key will be rotated every 6 months for security. You must update the Chronicle forwarder for Windows on Docker image every 6 months.
- The minimum batch size for forwarder is now increased to 200KB for better performance.
- Data compression is now enabled by default. It reduces the network bandwidth consumption by 80%.
- Hot config loading is now supported and applies configuration changes within 5 minutes without the need to restart the forwarder.
- Automatic buffering handles spikes in incoming traffic by efficiently using available memory on the host system. This feature is optional.

<b>Feature</b>	<p>Exclusions for Curated Detections</p> <p>You can now configure exclusions to more finely tune the results of the Curated Detections provided by the Google Cloud Threat Intelligence (GCTI) team.</p>	<p><a href="https://cloud.google.com/chronicle/docs/detection/rule-exclusions">https://cloud.google.com/chronicle/docs/detection/rule-exclusions</a></p> <p>Proprietary + Confidential</p>
<b>Feature</b>	<p>UDM Search Pivot Table</p> <p>The UDM Search Pivot Table enables you to further analyze your UDM search results, giving you the following capabilities:</p> <ul style="list-style-type: none"><li>● Group search results by up to five UDM fields.</li><li>● Perform aggregations (sum, count, count distinct, average, stddev, min, and max) on up to five values within the UDM fields (for example, domains, users, and products).</li><li>● Sort results of the pivot table (ascending, descending)</li></ul> <p>This feature is being enabled for global customers in a phased manner and is expected to fully roll out over the next month.</p>	<p><a href="https://cloud.google.com/chronicle/docs/investigation/udm-search#use_the_pivot_table_to_analyze_events">https://cloud.google.com/chronicle/docs/investigation/udm-search#use_the_pivot_table_to_analyze_events</a></p>
<b>Feature</b>	<p>Chronicle released the following additional data enrichment and precomputed analytic capabilities that can provide additional context during an investigation:</p> <ul style="list-style-type: none"><li>● Enriched entities with WHOIS data.</li><li>● Enriched entities with VirusTotal relationship data.</li><li>● Enriched events with VirusTotal file metadata.</li><li>● Data from Google Cloud Threat Intelligence curated threat feeds.</li><li>● Precomputed first-seen and last-seen occurrence for domains, IP addresses, and file hashes (SHA256, SHA1, MD5).</li><li>● Precomputed first-seen occurrence for assets and users.</li></ul>	<p><a href="https://cloud.google.com/chronicle/docs/event-processing/data-enrichment">https://cloud.google.com/chronicle/docs/event-processing/data-enrichment</a></p> <p><a href="https://cloud.google.com/chronicle/docs/investigation/use-enriched-data-in-search">https://cloud.google.com/chronicle/docs/investigation/use-enriched-data-in-search</a></p> <p><a href="https://cloud.google.com/chronicle/docs/detection/use-enriched-data-in-rules">https://cloud.google.com/chronicle/docs/detection/use-enriched-data-in-rules</a></p> <p>Google Cloud</p>

# Mise en place

Ingredients needed.

1. List of rules to start with. <https://github.com/chronicle/detection-rules>
2. GCP project to host your rules, reference lists, etc.
3. Non-Ingest API key for your Partner NFR Chronicle SIEM instance.
4. Healthy Python environment. Local to your machine or via tools like Colab.  
<https://collab.research.google.com>

## Suggested pairings

1. A list of regions or security zones
2. A list of forwarders including their configurations

# Elements of Advanced Rules

NOTE: Most of these concepts can be tested in Warstory for operating examples.

- Namespaces, Labels and Reference Lists for Security Zones
- Multiple Event Syntax
- Entity Data Triggering
- Outcome Calculation

# What are namespaces and labels

## Namespace

A namespace is a metadata attribute that identifies logs from distinct network segments. In addition to creating a metadata key, it also segments IP schemes such that the same IP range can exist in multiple log streams and they can be operated on by different rules.

## Label

A Label is an arbitrary metadata attribute that can be used in rules, but they do not affect the entity graph in the same manner as namespaces.

# How do namespaces and labels affect rules?






Define those zones! PCI? OT? Purdue Model Layer? AD Region?

Forwarder ID and Forwarder Name in UDM?

# Namespace example

```
metadata:
  namespace: FORWARDER
collectors:
- syslog:
  common:
    metadata:
      namespace: CORPORATE
      batch_n_bytes: 1048576
      batch_n_seconds: 10
      data_hint: null
      data_type: NIX_SYSTEM
      enabled: true
    tcp_address: 0.0.0.0:30000
    connection_timeout_sec: 60
- syslog:
  common:
    batch_n_bytes: 1048576
    batch_n_seconds: 10
    data_hint: null
    data_type: WINEVTLOG
    enabled: true
    tcp_address: 0.0.0.0:30001
    connection_timeout_sec: 60
```

## In the Search UI

Q 10.0.0.1 <span>⊗</span>	
ASSETS	
	[untagged] > 10.0.0.1
	aws-account1:vpc > 10.0.0.1
	black-mesa-labs:vpc1 > 10.0.0.1
	xyzenterprise.com > 10.0.0.1
	[all namespaces] > 10.0.0.1

## In Asset View

10.0.0.1

SEARCH

2 HOURS

ASSET

black-mesa-labs:vpc1 > user.corp.google.com

IP Addresses: [100.106.164.22](#) MAC Addresses: [84:fe:z5:1d:b9:4m](#) Environment: GCP Labels: `enableIntegrityMonitoring: true` Project Name: `ab-12345-acme-xy1`

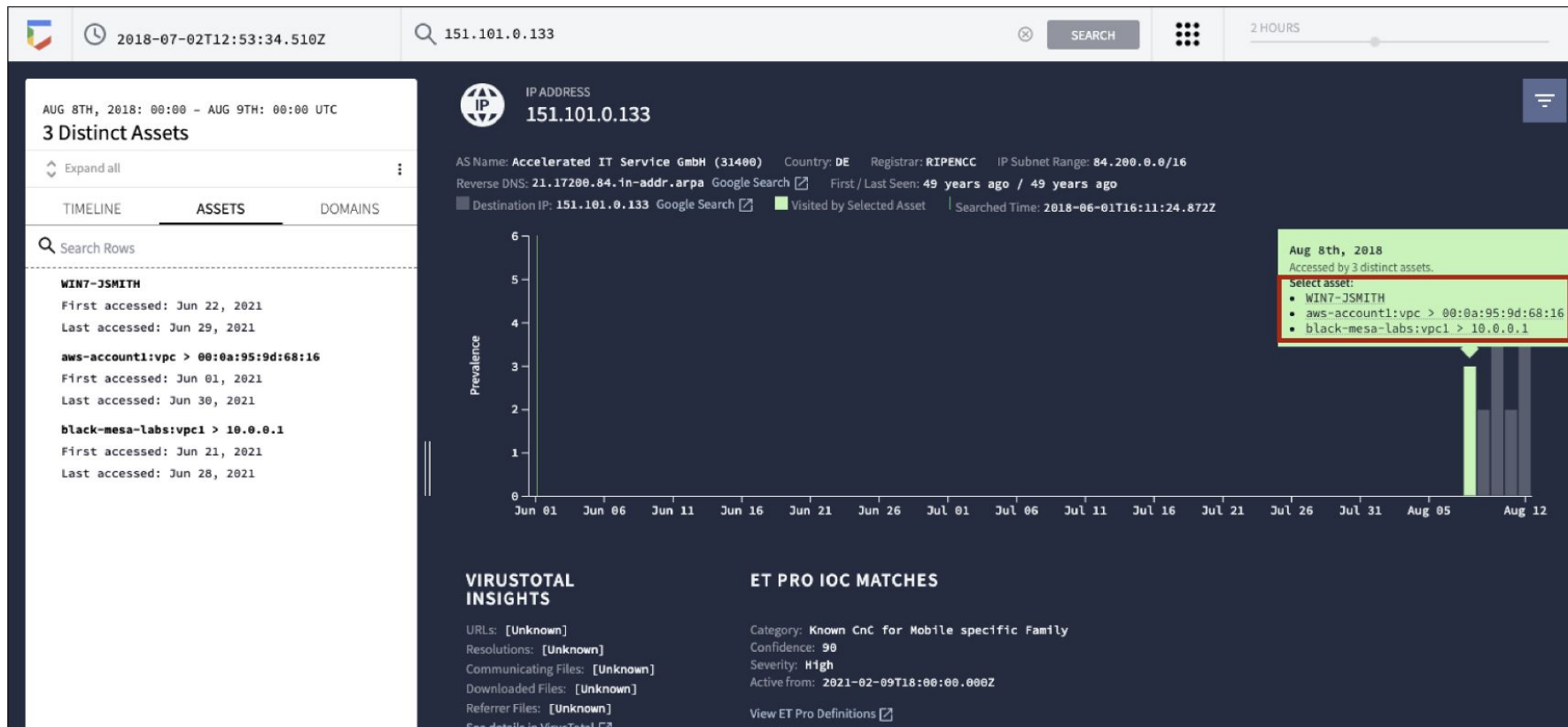
Location: `europa-west3` Zone: `europa-west3-a` Hardware Model: `e2-standard-4` CPU Platform: `Intel Broadwell` Platform: `WINDOWS`

Platform Version: `Debian GNU/Linux 10 (buster)` Platform Patch Level: `#1 SMP Debian 4.19.160-2` Resource Created: `9 months ago` Asset ID: `asset-id-123`

First observed: `22 days ago` Data last collected: `22 days ago`

Selected Event Selected domain Domain First/Recurring 2 alerts | Searched time: 08:00:00

# In IP Address, Domain and Hash Views



# Label configuration example

```
metadata:  
  labels:  
    foo: bar  
    meow: mix  
collectors:  
syslog:  
  common:  
    metadata:  
      labels:  
        foo: baz  
        meow: mix  
    batch_n_bytes: 1048576  
    batch_n_seconds: 10  
    data_hint: null  
    data_type: NIX_SYSTEM  
    enabled: true  
    tcp_address: 0.0.0.0:30000  
    connection_timeout_sec: 60
```

Find it in UDM Search?

```
additional.fields["pod_name"] = "kube-scheduler"  
metadata.ingestion_labels["MetadataKeyDeletion"] = "startup-script"
```

# Use labels and namespaces in Rules?

events:

```
// Network Connection Event
```

```
$e.metadata.event_type = "NETWORK_CONNECTION"
```

```
$e.target.ip = $tor_ip
```

```
$e.metadata.namespace = "PCI"
```

```
$e.metadata.ingestion_labels_key = "foo"
```

```
$e.metadata.ingestion_labels_value = "baz"
```

```
// $e.metadata.ingestion_label["foo"] = baz
```

```
// Tor IP search in GCTI Feed
```

```
$tor.graph.entity.artifact.ip = $tor_ip
```

```
$tor.graph.metadata.entity_type = "IP_ADDRESS"
```

```
$tor.graph.metadata.threat.threat_feed_name = "Tor Exit Nodes"
```

```
$tor.graph.metadata.source_type = "GLOBAL_CONTEXT"
```

# Reference Lists

A reference list is a generic list of values which can be used to analyze your data.

<https://cloud.google.com/chronicle/docs/reference/reference-lists>

Create lists that can be used in conjunction with rules

- Duplicate or add to existing lists

## Rule Syntax

```
field/variables in %list_name
```

```
field/variables in cidr %list_name
```

```
field/variables in regex %list_name
```

## Examples

- `$selection1.principal.hostname in %key_servers`
- `re.capture($selection1.target.process.file.full_path, /.*\((.*)/) in %processes_of_interest`

# Reference List Examples

NOTE: A Reference maps to one key only.

## aws\_accounts

127632175811

177632175808

## key\_server\_labels

activedir

sql

exchange

sap

## sensitive\_namespaces

PCI

ITAR

HIPAA

# Entity Data

- Provides additional contextualization to the events
- A good bit of contextualization is absorbed into the event stream natively without adding the entity data model fields directly (Enrichment)
- Use Case
  - A rule could be run limiting the scope to just those in the Finance department based on LDAP data of the users
- Entity data is associated with the event at the time the event occurred
  - Example - Alice works in the Finance department until April 30 but on May 1 she moved to the Fraud department and her LDAP data (source of Entity data) is updated
  - Rules evaluating the Finance department activities would only apply to her during her time in Finance (up until April 30)

# Rule Syntax

```
rule mitre_attack_T1021_002_windows_admin_share_with_user_enrichment {  
  meta:  
  
  events:  
    $event.target.process.command_line = /net.*use.*(C|ADMIN|IPC)\$/ nocase  
    $event.principal.user.userid = $userid  
  
    $event.user.department != "Information Technology" or  
    $event.principal.user.title = "Intern"  
  
  match:  
    $userid over 5m  
  
  condition:  
    $event  
}
```

# A brief interlude with Colab

Using APIs to make this happen.

# I have all of these ingredients? What now?

Putting it all together.

An Example Directory Hierarchy:

reflist - to contain the reference lists

rule-creator - location for scripts that push/compare/enable rules in Chronicle

rules - the rules as discrete yaral files

<https://github.com/chronicle/api-samples-python>

<https://github.com/chronicle/detection-rules>

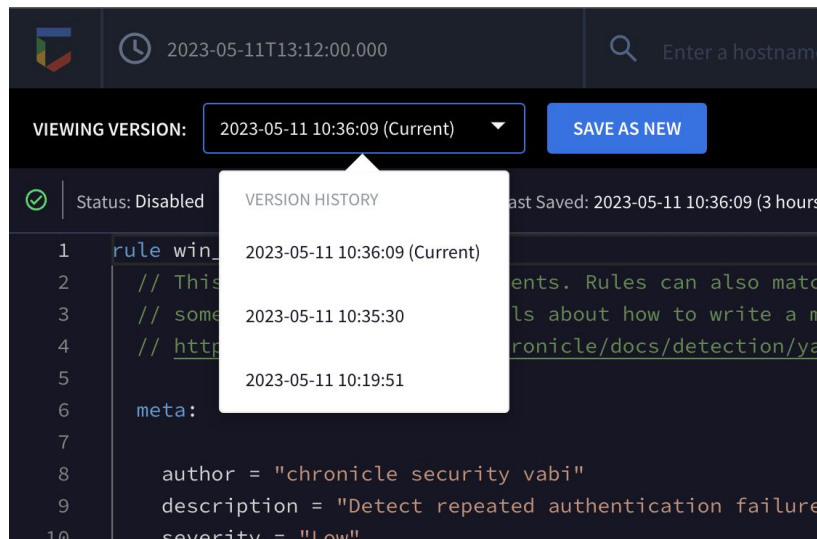
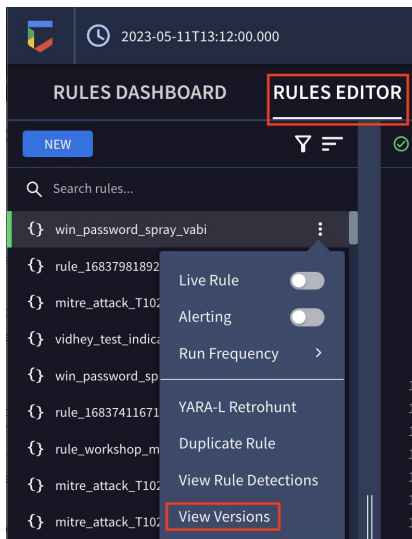
<https://github.com/chronicle/GCTI>

# I think I broke it.

How do I recover?

Repo destruction can happen. How can we help?

- Chronicle has a built in content versioning system.
- The API supports pulling and pushing different versions of content.
- You can enable a previous version of content without a local copy.



VIEWING VERSION: 2023-05-11 10:36:09 (Current)

SAVE AS NEW

VIEW DETECTIONS

COMPARE VERSIONS

EXIT

Status: Disabled Created: 2023-05-11 10:19:51 Last Saved: 2023-05-11 10:36:09 (3 hours ago) Rule Type: Multiple Events

## COMPARE TWO VERSIONS

CLOSE COMPARE

EXIT

COMPARE: 2023-05-11 10:19:51

CURRENT: 2023-05-11 10:36:09

SAVE AS NEW

```
1 rule win_password_spray_vabi {
2   // This rule matches single events. Rules can also match multiple events within
3   // some time window. For details about how to write a multi-event rule, see
4   // https://cloud.google.com/chronicle/docs/detection/yara-l-2-0-overview#single-event_v
5
6   meta:
7     // Allows for storage of arbitrary key-value pairs of rule details - who
8     // wrote it, what it detects on, version control, etc.
9     // The "author" and "severity" fields are special, as they are used as
10    // columns on the rules dashboard. If you'd like to be able to sort based on
11    // these fields on the dashboard, make sure to add them here.
12    // Severity value, by convention, should be "Low", "Medium" or "High"
13    author = "analyst123"
14    description = "8:00 AM local time"
15    severity = "Medium"
```

```
1 rule win_password_spray_vabi {
2   // This rule matches single events. Rules can also match multiple events within
3   // some time window. For details about how to write a multi-event rule, see
4   // https://cloud.google.com/chronicle/docs/detection/yara-l-2-0-overview#single-event_v
5
6   meta:
7+
8+   author = "chronicle security vabi"
9+   description = "Detect repeated authentication failure from the same host but with mul
10+  severity = "Low"
11+
12+  // FIELDS OF INTEREST
13+  //metadata.event_type - USER_LOGIN
14+  //metadata.vendor_name - Microsoft
15+  //security_result.action - BLOCK
16+  //same host failing while using numerous userids (more than 10) over a 30 minute time
17+  //test your rule over the past 3 days. -- execution
```

# A brief interlude with Colab

Using APIs to make this happen.



# Thank you



Google