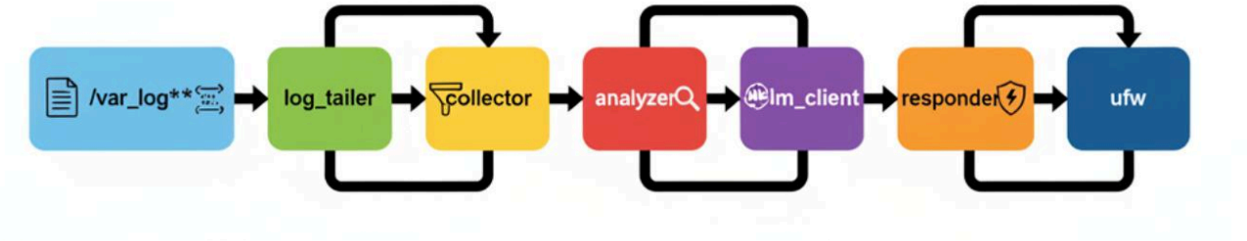
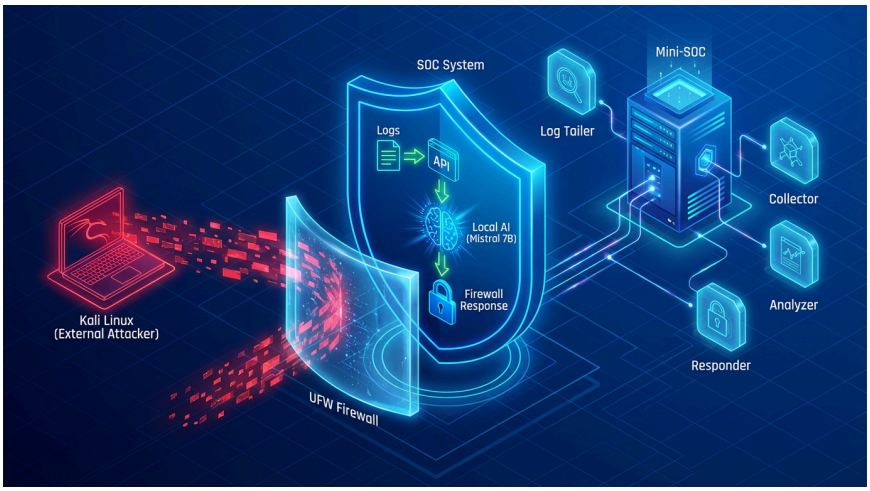
Building an AI-Powered Mini-SOC - Real Attack Detection & Response

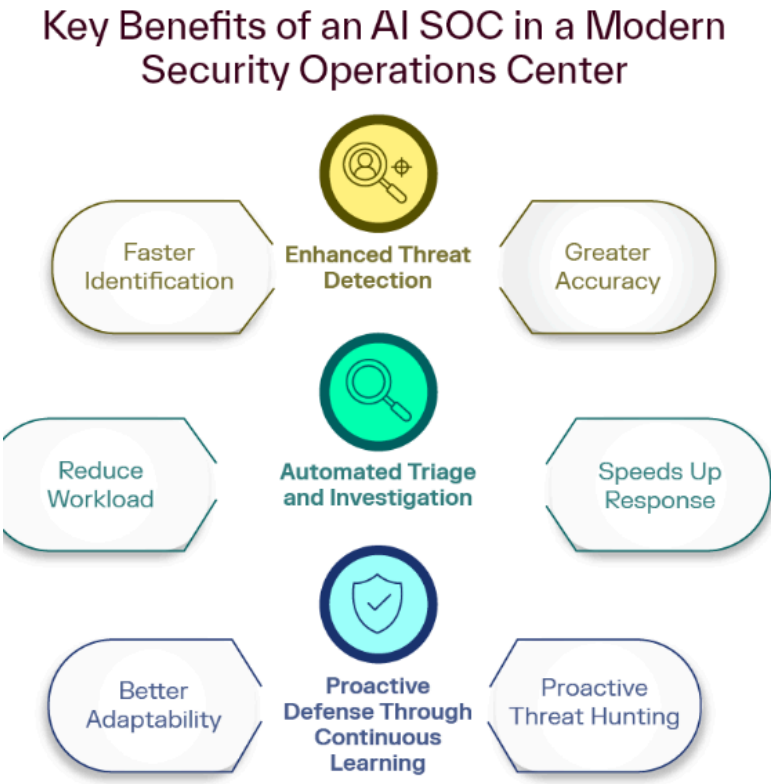
In cybersecurity, sending sensitive logs to the cloud for analysis is often a privacy risk. I wanted to solve this by building a **Mini-SOC** that keeps all data on-premise while still using the power of Generative AI.





The Challenge:
How to detect and block attacks like SSH Brute-force and Network Scanning in real-time, without relying on external APIs or human intervention?

The Solution:
I engineered a custom Python-based pipeline that integrates **Mistral 7B** (running locally via LM Studio) directly into the defense loop.



🛠️ System Specs:

- **Eyes:** A `Log Tailer` agent watching `/var/log` in real-time.
- **Brain:** A Hybrid Analyzer. It uses fast heuristics for speed and a Local LLM for deep context analysis.
- **Muscle:** A `Responder` agent that autonomously updates `UFW` firewall rules to ban malicious IPs

```
(venv) chaimae@ubuntuCHA:~/soc-project$ python3 responder.py

      RESPONDER DÉMARRÉ
      Port: 6003
      Mode: SIMULATION

 * Serving Flask app 'responder'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:6003
 * Running on http://192.168.11.102:6003
Press CTRL+C to quit
```

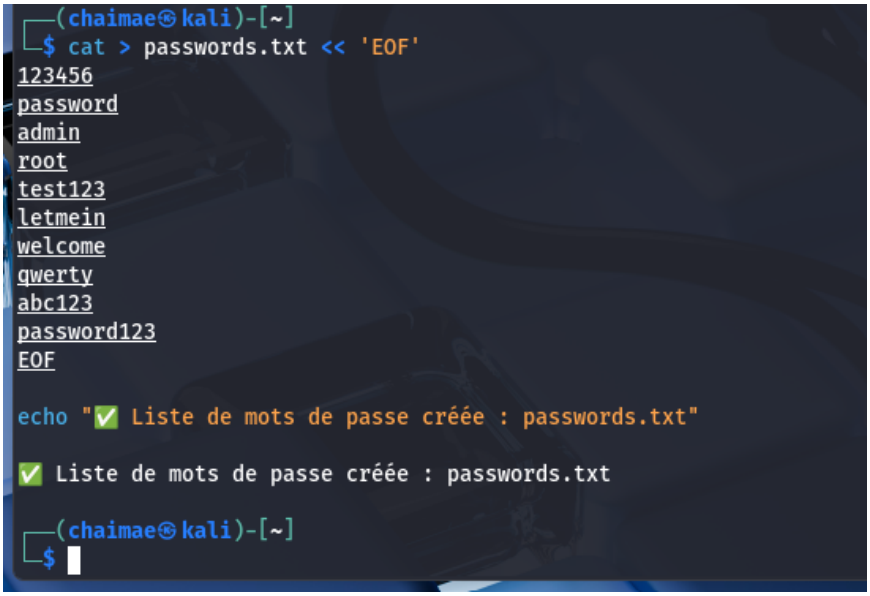
```
(venv) chaimae@ubuntuCHA:~/soc-project$ python3 analyzer.py

      ANALYZER DÉMARRÉ
      Port: 6002
      LM Studio: http://127.0.0.1:1234

[ANALYZER] ✅ LM Studio connecté
 * Serving Flask app 'analyzer'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:6002
 * Running on http://192.168.11.102:6002
Press CTRL+C to quit
```

```
(venv) chaimae@ubuntuCHA:~/soc-project$ python3 collector.py

      COLLECTOR DÉMARRÉ
      Port: 6001
      Prêt à recevoir événements

 * Serving Flask app 'collector'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:6001
 * Running on http://192.168.11.102:6001
Press CTRL+C to quit
```

```
 GNU nano 7.2                                                    log_tailer.py
import sys
import config

def tail_file(filepath):
    """Générateur qui lit un fichier en continu (comme tail -f)"""
    try:
        with open(filepath, 'r') as f:
            # Se positionner à la fin du fichier
            f.seek(0, 2)
            while True:
                line = f.readline()
                if line:
                    yield line.strip()
                else:
                    time.sleep(0.1)
    except FileNotFoundError:
        print(f"[Tailer] ⚠️Fichier non trouvé : {filepath}")
    except PermissionError:
        print(f"[Tailer] ❌ Permission refusée : {filepath} (exécuter avec sudo)")


def parse_ssh_log(line):
    """Parse une ligne de /var/log/auth.log"""

    # Pattern : Failed password for USER from IP port PORT
    pattern = r'Failed password for (\w+) from ([\d\.]+) port (\d+)'
    match = re.search(pattern, line)

    if match:
        return {
            "id": f"ssh-{int(time.time() * 1000)}",
            "ts": datetime.datetime.now().isoformat(),
            "kind": "ssh_failed",
            "src_ip": match.group(2),
            "dst": "ubuntu-soc",
            "user": match.group(1),
```

⚔️ ATTACK 1: SSH Brute-Force with Hydra

Create a password list.



Hydra attack

 # SSH brute-force attack

🌐 IP Addresses

- **Kali VM IP Address:**
  - the IPv4 address: `192.168.11.123`
- **Ubuntu VM IP Address (Target/SOC):**
  - the IPv4 address: `192.168.11.126`

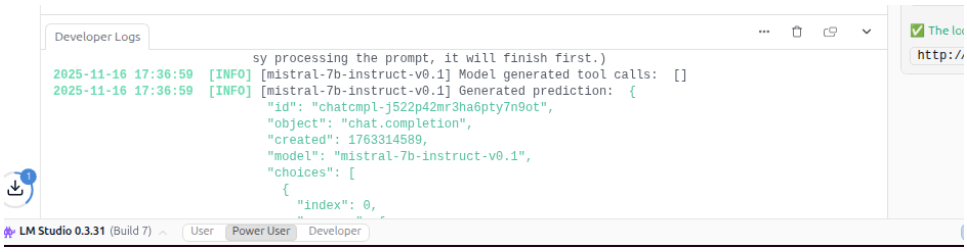
TARGET_IP = “192.168.11.126”

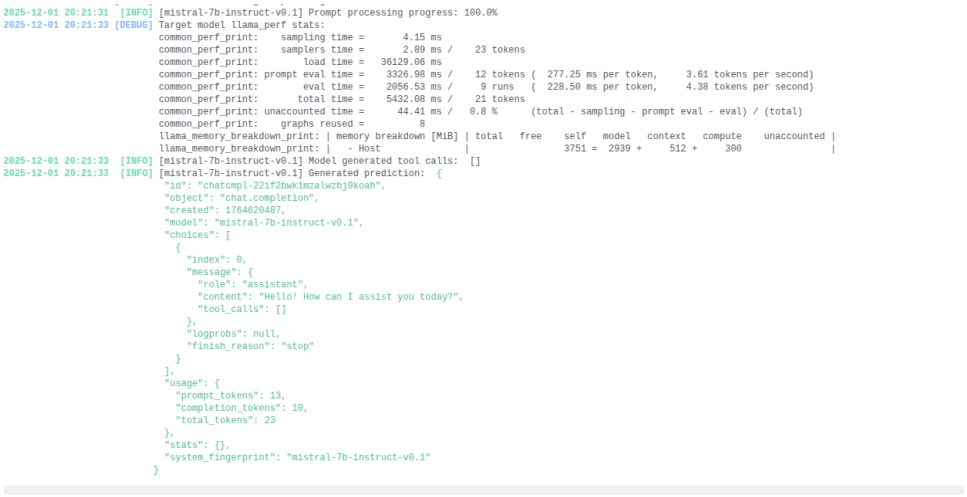
```
┌──(chaimae㉿kali)-[~]
└─$ export TARGET_IP="192.168.11.126"

┌──(chaimae㉿kali)-[~]
└─$ ping 192.168.11.126
PING 192.168.11.126 (192.168.11.126) 56(84) bytes of data.
64 bytes from 192.168.11.126: icmp_seq=1 ttl=64 time=11.8 ms
64 bytes from 192.168.11.126: icmp_seq=2 ttl=64 time=1.84 ms
64 bytes from 192.168.11.126: icmp_seq=3 ttl=64 time=1.23 ms
64 bytes from 192.168.11.126: icmp_seq=4 ttl=64 time=1.04 ms
^C
--- 192.168.11.126 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3051ms
rtt min/avg/max/mdev = 1.037/3.969/11.780/4.518 ms
```

```
┌──(chaimae㉿kali)-[~]
└─$ hydra -l chaimae -P ~/passwords.txt ssh://$TARGET_IP -t 4 -V
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal p
urposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-12-01 05:28:40
[DATA] max 4 tasks per 1 server, overall 4 tasks, 10 login tries (l:1/p:10), ~3 tries per task
[DATA] attacking ssh://192.168.11.126:22/
[ATTEMPT] target 192.168.11.126 - login "chaimae" - pass "123456" - 1 of 10 [child 0] (0/0)
[ATTEMPT] target 192.168.11.126 - login "chaimae" - pass "password" - 2 of 10 [child 1] (0/0)
[ATTEMPT] target 192.168.11.126 - login "chaimae" - pass "admin" - 3 of 10 [child 2] (0/0)
[ATTEMPT] target 192.168.11.126 - login "chaimae" - pass "root" - 4 of 10 [child 3] (0/0)
[ATTEMPT] target 192.168.11.126 - login "chaimae" - pass "test123" - 5 of 10 [child 3] (0/0)
[ATTEMPT] target 192.168.11.126 - login "chaimae" - pass "letmein" - 6 of 10 [child 0] (0/0)
[ATTEMPT] target 192.168.11.126 - login "chaimae" - pass "welcome" - 7 of 10 [child 1] (0/0)
[ATTEMPT] target 192.168.11.126 - login "chaimae" - pass "qwerty" - 8 of 10 [child 2] (0/0)
[ATTEMPT] target 192.168.11.126 - login "chaimae" - pass "abc123" - 9 of 10 [child 3] (0/0)
[ATTEMPT] target 192.168.11.126 - login "chaimae" - pass "password123" - 10 of 10 [child 1] (0/0)
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-12-01 05:28:50

┌──(chaimae㉿kali)-[~]
```

└─$ hydra -l chaimae -p passw0rdFaux ssh://100.102.241.131 -t 16

hydra -l chaimae -P ~/passwords.txt ssh://$TARGET_IP -t 4 -V






```
chaimae@ubuntuCHA:~/soc-project$ sudo cat /var/log/auth.log
[sudo] password for chaimae:
2025-11-16T14:16:47.417699+00:00 ubuntuCHA gdm-launch-environment]: pam_unix(
2025-11-16T14:16:47.439510+00:00 ubuntuCHA systemd-logind[891]: New session c
```

Enable real Block

```
# ═══════════════════════════════════════
# 3. MODE D'OPÉRATION
# ═══════════════════════════════════════
DRY_RUN = False # True = Simulation, False = Blocage réel

# ═══════════════════════════════════════
# 4. BACKEND DE BLOCAGE
# ═══════════════════════════════════════
BLOCKING_BACKEND = "ufw"  # Options: "ufw" ou "iptables"
```

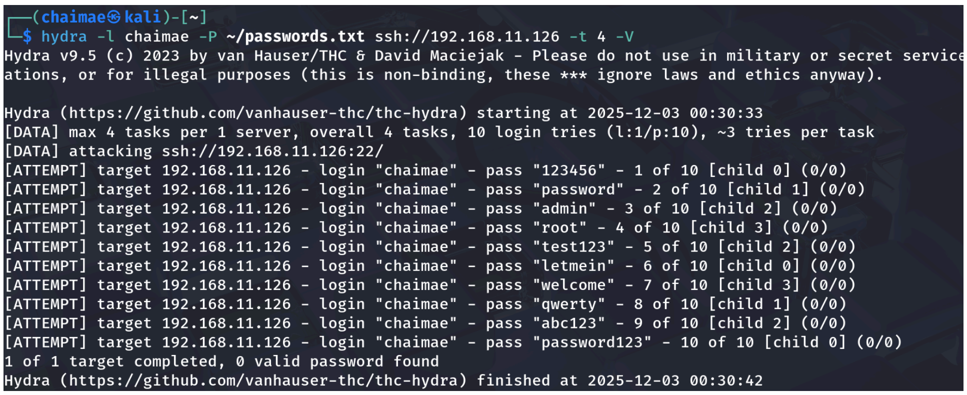
🚀 **The Result:**
The system successfully detected a simulated Hydra attack and blocked the intruder's IP within seconds. The best part? It cost $0 in cloud fees and zero data left my local network.

./LM-Studio-0.3.31-7-x64.AppImage --no-sandbox

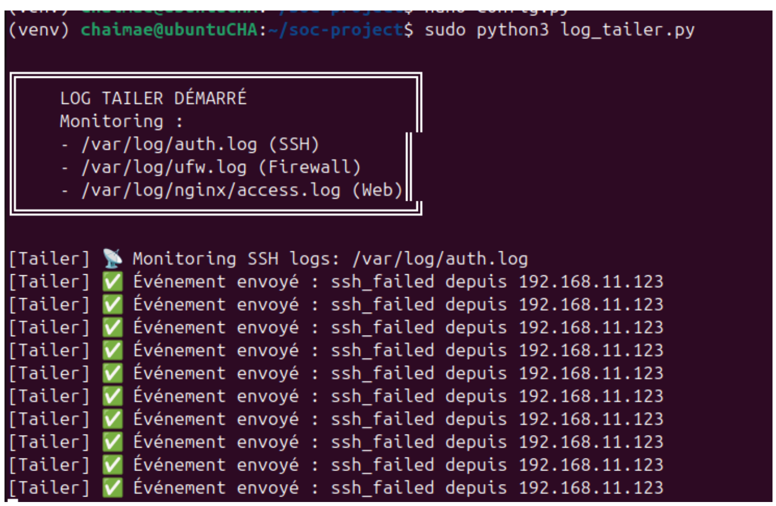
Hydra attempts SSH connections

- Hydra sends 10 SSH login attempts with different passwords

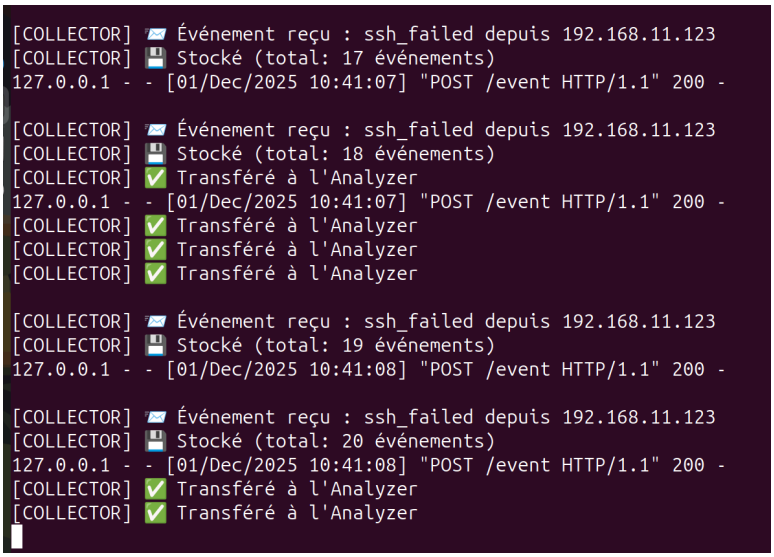
- Each attempt fails (authentication failure)
- 2️⃣ Ubuntu logs the failures in `/var/log/auth.log`
- 3️⃣ The Log Tailer reads these lines
- 5️⃣ The Analyzer counts the failures
- 6️⃣ The Responder blocks the IP (if not in the whitelist)

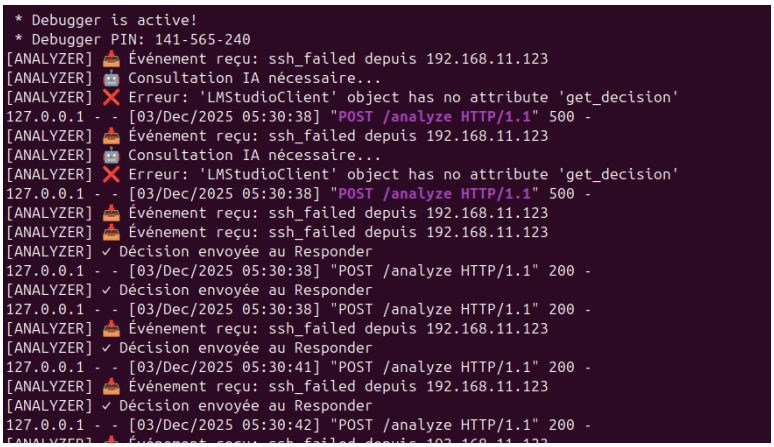


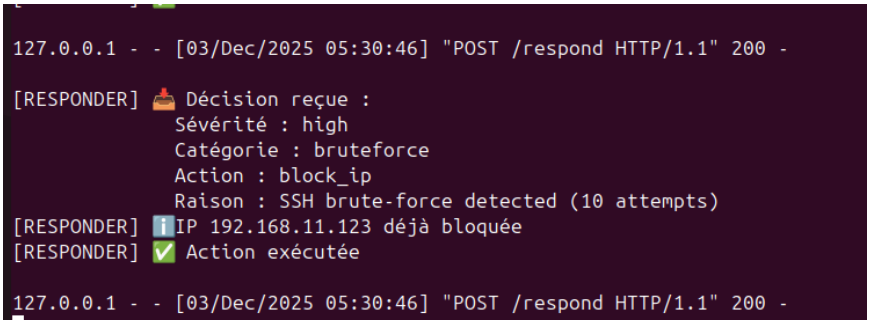
```
┌──(chaimae㉿kali)-[~]
└─$ hydra -l chaimae -P ~/passwords.txt ssh://192.168.11.126 -t 4 -V
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service
ations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-12-03 00:30:33
[DATA] max 4 tasks per 1 server, overall 4 tasks, 10 login tries (l:1/p:10), ~3 tries per task
[DATA] attacking ssh://192.168.11.126:22/
[ATTEMPT] target 192.168.11.126 - login "chaimae" - pass "123456" - 1 of 10 [child 0] (0/0)
[ATTEMPT] target 192.168.11.126 - login "chaimae" - pass "password" - 2 of 10 [child 1] (0/0)
[ATTEMPT] target 192.168.11.126 - login "chaimae" - pass "admin" - 3 of 10 [child 2] (0/0)
[ATTEMPT] target 192.168.11.126 - login "chaimae" - pass "root" - 4 of 10 [child 3] (0/0)
[ATTEMPT] target 192.168.11.126 - login "chaimae" - pass "test123" - 5 of 10 [child 2] (0/0)
[ATTEMPT] target 192.168.11.126 - login "chaimae" - pass "letmein" - 6 of 10 [child 0] (0/0)
[ATTEMPT] target 192.168.11.126 - login "chaimae" - pass "welcome" - 7 of 10 [child 3] (0/0)
[ATTEMPT] target 192.168.11.126 - login "chaimae" - pass "qwerty" - 8 of 10 [child 1] (0/0)
[ATTEMPT] target 192.168.11.126 - login "chaimae" - pass "abc123" - 9 of 10 [child 2] (0/0)
[ATTEMPT] target 192.168.11.126 - login "chaimae" - pass "password123" - 10 of 10 [child 0] (0/0)
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-12-03 00:30:42
```

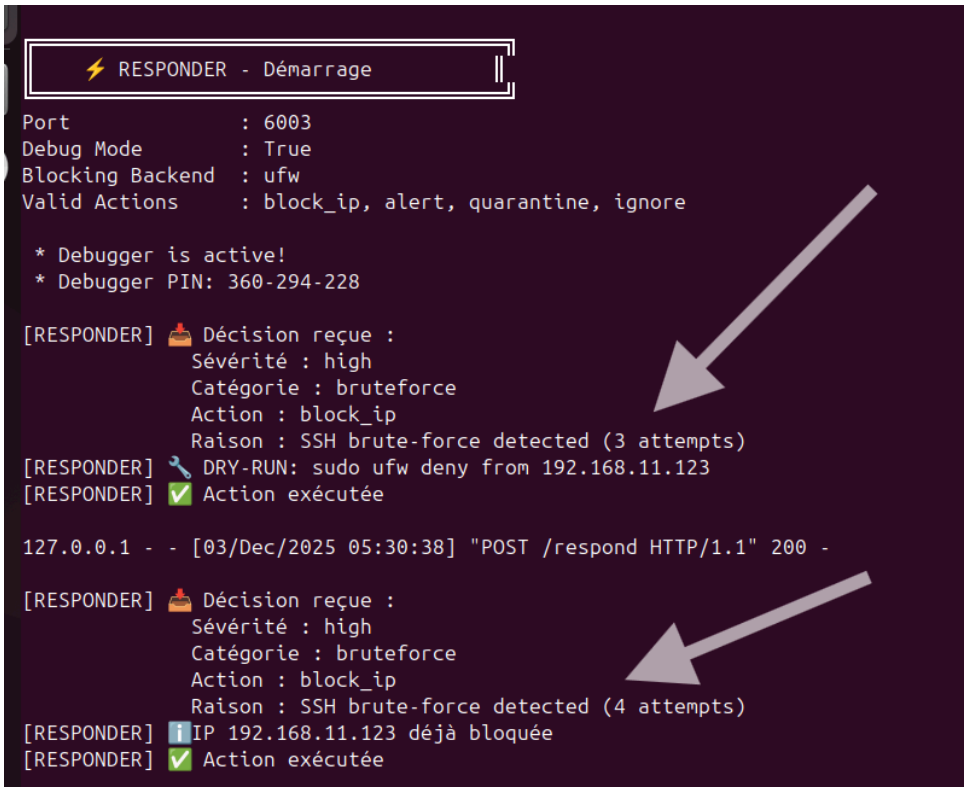


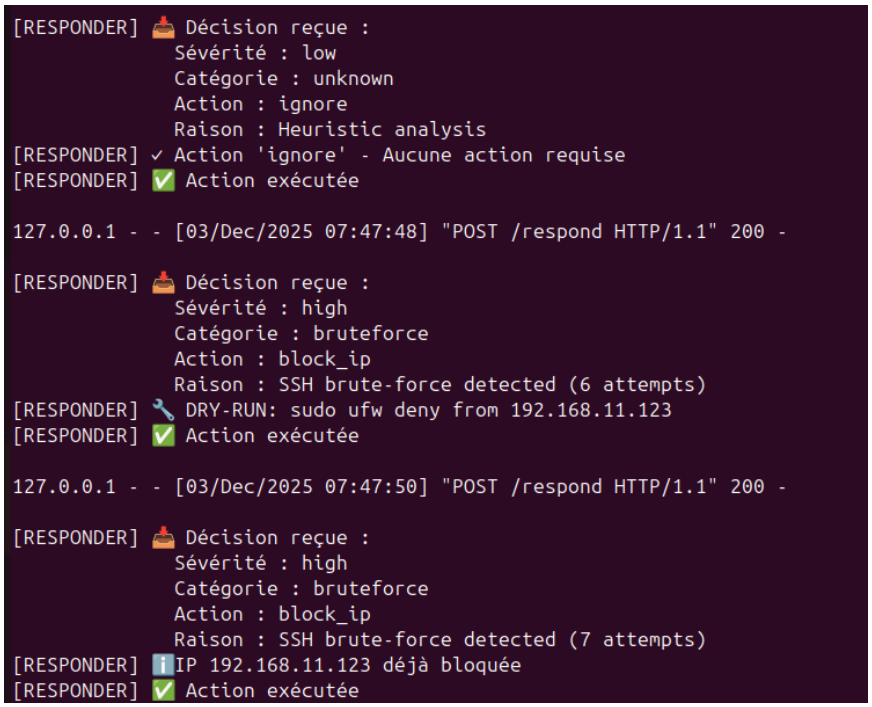
```
(venv) chaimae@ubuntuCHA:~/soc-project$ sudo python3 log_tailer.py

    LOG TAILER DÉMARRÉ
    Monitoring :
    - /var/log/auth.log (SSH)
    - /var/log/ufw.log (Firewall)
    - /var/log/nginx/access.log (Web)

[Tailer] 📡 Monitoring SSH logs: /var/log/auth.log
[Tailer] ✅ Événement envoyé : ssh_failed depuis 192.168.11.123
[Tailer] ✅ Événement envoyé : ssh_failed depuis 192.168.11.123
[Tailer] ✅ Événement envoyé : ssh_failed depuis 192.168.11.123
[Tailer] ✅ Événement envoyé : ssh_failed depuis 192.168.11.123
[Tailer] ✅ Événement envoyé : ssh_failed depuis 192.168.11.123
[Tailer] ✅ Événement envoyé : ssh_failed depuis 192.168.11.123
[Tailer] ✅ Événement envoyé : ssh_failed depuis 192.168.11.123
[Tailer] ✅ Événement envoyé : ssh_failed depuis 192.168.11.123
[Tailer] ✅ Événement envoyé : ssh_failed depuis 192.168.11.123
[Tailer] ✅ Événement envoyé : ssh_failed depuis 192.168.11.123
```


[COLLECTOR] 📨 Événement reçu : ssh_failed depuis 192.168.11.123
[COLLECTOR] 💾 Stocké (total: 17 événements)
127.0.0.1 - - [01/Dec/2025 10:41:07] "POST /event HTTP/1.1" 200 -
[COLLECTOR] 📨 Événement reçu : ssh_failed depuis 192.168.11.123
[COLLECTOR] 💾 Stocké (total: 18 événements)
[COLLECTOR] ✅ Transféré à l'Analyzer
127.0.0.1 - - [01/Dec/2025 10:41:07] "POST /event HTTP/1.1" 200 -
[COLLECTOR] ✅ Transféré à l'Analyzer
[COLLECTOR] ✅ Transféré à l'Analyzer
[COLLECTOR] ✅ Transféré à l'Analyzer
[COLLECTOR] 📨 Événement reçu : ssh_failed depuis 192.168.11.123
[COLLECTOR] 💾 Stocké (total: 19 événements)
127.0.0.1 - - [01/Dec/2025 10:41:08] "POST /event HTTP/1.1" 200 -
[COLLECTOR] 📨 Événement reçu : ssh_failed depuis 192.168.11.123
[COLLECTOR] 💾 Stocké (total: 20 événements)
127.0.0.1 - - [01/Dec/2025 10:41:08] "POST /event HTTP/1.1" 200 -
[COLLECTOR] ✅ Transféré à l'Analyzer
[COLLECTOR] ✅ Transféré à l'Analyzer


* Debugger is active!
* Debugger PIN: 141-565-240
[ANALYZER] 📥 Événement reçu: ssh_failed depuis 192.168.11.123
[ANALYZER] 🤖 Consultation IA nécessaire...
[ANALYZER] ❌ Erreur: 'LMStudioClient' object has no attribute 'get_decision'
127.0.0.1 - - [03/Dec/2025 05:30:38] "POST /analyze HTTP/1.1" 500 -
[ANALYZER] 📥 Événement reçu: ssh_failed depuis 192.168.11.123
[ANALYZER] 🤖 Consultation IA nécessaire...
[ANALYZER] ❌ Erreur: 'LMStudioClient' object has no attribute 'get_decision'
127.0.0.1 - - [03/Dec/2025 05:30:38] "POST /analyze HTTP/1.1" 500 -
[ANALYZER] 📥 Événement reçu: ssh_failed depuis 192.168.11.123
[ANALYZER] 📥 Événement reçu: ssh_failed depuis 192.168.11.123
[ANALYZER] ✓ Décision envoyée au Responder
127.0.0.1 - - [03/Dec/2025 05:30:38] "POST /analyze HTTP/1.1" 200 -
[ANALYZER] ✓ Décision envoyée au Responder
127.0.0.1 - - [03/Dec/2025 05:30:38] "POST /analyze HTTP/1.1" 200 -
[ANALYZER] 📥 Événement reçu: ssh_failed depuis 192.168.11.123
[ANALYZER] ✓ Décision envoyée au Responder
127.0.0.1 - - [03/Dec/2025 05:30:41] "POST /analyze HTTP/1.1" 200 -
[ANALYZER] 📥 Événement reçu: ssh_failed depuis 192.168.11.123
[ANALYZER] ✓ Décision envoyée au Responder
127.0.0.1 - - [03/Dec/2025 05:30:42] "POST /analyze HTTP/1.1" 200 -


127.0.0.1 - - [03/Dec/2025 05:30:46] "POST /respond HTTP/1.1" 200 -
[RESPONDER] 📥 Décision reçue :
Sévérité : high
Catégorie : bruteforce
Action : block_ip
Raison : SSH brute-force detected (10 attempts)
[RESPONDER] ℹ️IP 192.168.11.123 déjà bloquée
[RESPONDER] ✅ Action exécutée
127.0.0.1 - - [03/Dec/2025 05:30:46] "POST /respond HTTP/1.1" 200 -


⚡ RESPONDER - Démarrage
Port : 6003
Debug Mode : True
Blocking Backend : ufw
Valid Actions : block_ip, alert, quarantine, ignore
* Debugger is active!
* Debugger PIN: 360-294-228
[RESPONDER] 📥 Décision reçue :
Sévérité : high
Catégorie : bruteforce
Action : block_ip
Raison : SSH brute-force detected (3 attempts)
[RESPONDER] 🔧 DRY-RUN: sudo ufw deny from 192.168.11.123
[RESPONDER] ✅ Action exécutée
127.0.0.1 - - [03/Dec/2025 05:30:38] "POST /respond HTTP/1.1" 200 -
[RESPONDER] 📥 Décision reçue :
Sévérité : high
Catégorie : bruteforce
Action : block_ip
Raison : SSH brute-force detected (4 attempts)
[RESPONDER] ℹ️IP 192.168.11.123 déjà bloquée
[RESPONDER] ✅ Action exécutée


[RESPONDER] 📥 Décision reçue :
Sévérité : low
Catégorie : unknown
Action : ignore
Raison : Heuristic analysis
[RESPONDER] ✓ Action 'ignore' - Aucune action requise
[RESPONDER] ✅ Action exécutée
127.0.0.1 - - [03/Dec/2025 07:47:48] "POST /respond HTTP/1.1" 200 -
[RESPONDER] 📥 Décision reçue :
Sévérité : high
Catégorie : bruteforce
Action : block_ip
Raison : SSH brute-force detected (6 attempts)
[RESPONDER] 🔧 DRY-RUN: sudo ufw deny from 192.168.11.123
[RESPONDER] ✅ Action exécutée
127.0.0.1 - - [03/Dec/2025 07:47:50] "POST /respond HTTP/1.1" 200 -
[RESPONDER] 📥 Décision reçue :
Sévérité : high
Catégorie : bruteforce
Action : block_ip
Raison : SSH brute-force detected (7 attempts)
[RESPONDER] ℹ️IP 192.168.11.123 déjà bloquée
[RESPONDER] ✅ Action exécutée

```
chaimae@ubuntuCHA:~/soc-project$ python3 lm_client.py
=== Test LM Studio Client ===

Test 1: Connexion... ✓ OK

Test 2: Analyse événement...
[LM_CLIENT] 📤 Envoi à LM Studio: http://localhost:1234/v1/chat/completions
[LM_CLIENT] 📋 Event: ssh_failed from 192.168.11.126
```


```
common_perf_print:    graphs reused =         10
llama_memory_breakdown_print: | memory breakdown [MiB] | total   free    self   model   c
llama_memory_breakdown_print: |   - Host               |                 3751 =  2939 +
2025-12-03 07:53:57  [INFO] [mistral-7b-instruct-v0.1] Model generated tool calls:  []
2025-12-03 07:53:57  [INFO] [mistral-7b-instruct-v0.1] Generated prediction:  {
 "id": "chatcmpl-f5g8mvp7cfbp0g2v663at",
 "object": "chat.completion",
 "created": 1764748373,
 "model": "mistral-7b-instruct-v0.1",
 "choices": [
   {
     "index": 0,
     "message": {
       "role": "assistant",
       "content": "{\n  \"severity\": \"medium\",",
       "tool_calls": []
     },
     "logprobs": null,
     "finish_reason": "stop"
   }
 ],
 "usage": {
   "prompt_tokens": 208,
   "completion_tokens": 12,
   "total_tokens": 220
 },
 "stats": {},
 "system_fingerprint": "mistral-7b-instruct-v0.1"
}
```


```
┌──(chaimae㉿kali)-[~]
└─$ nmap -p 1-10 192.168.11.126 --max-rate 5 -v

Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-03 03:06 EST
Initiating ARP Ping Scan at 03:06
Scanning 192.168.11.126 [1 port]
Completed ARP Ping Scan at 03:06, 0.09s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 03:06
Completed Parallel DNS resolution of 1 host. at 03:06, 0.07s elapsed
Initiating SYN Stealth Scan at 03:06
Scanning 192.168.11.126 (192.168.11.126) [10 ports]
Completed SYN Stealth Scan at 03:07, 4.01s elapsed (10 total ports)
Nmap scan report for 192.168.11.126 (192.168.11.126)
Host is up (0.0015s latency).

PORT   STATE    SERVICE
1/tcp  filtered tcpmux
2/tcp  filtered compressnet
3/tcp  filtered compressnet
4/tcp  filtered unknown
5/tcp  filtered rje
6/tcp  filtered unknown
7/tcp  filtered echo
```

🎯 Gobuster test :

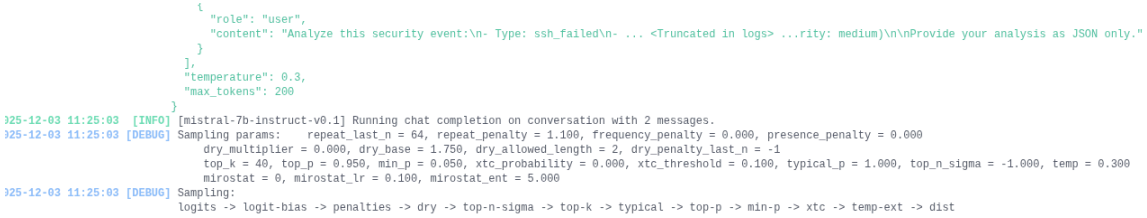
```
┌──(chaimae㉿kali)-[~]
└─$ # Sur Kali
cat > ~/test_web.txt << EOF
admin
login
config
backup
test
secret
dashboard
api
.git
.env
wp-admin
phpmyadmin
uploads
images
css
js
EOF
```

```
(venv) chaimae@ubuntuCHA:~/soc-project$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
     Active: active (running) since Wed 2025-12-03 09:11:07 UTC; 5min ago
       Docs: man:nginx(8)
   Main PID: 32636 (nginx)
      Tasks: 7 (limit: 6973)
     Memory: 5.1M (peak: 11.2M)
        CPU: 132ms
     CGroup: /system.slice/nginx.service
             ├─32636 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             ├─32639 "nginx: worker process"
             ├─32640 "nginx: worker process"
             ├─32641 "nginx: worker process"
             ├─32642 "nginx: worker process"
             ├─32643 "nginx: worker process"
             └─32644 "nginx: worker process"

Dec 03 09:11:07 ubuntuCHA systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
Dec 03 09:11:07 ubuntuCHA systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
(venv) chaimae@ubuntuCHA:~/soc-project$
```

```
┌──(chaimae㉿kali)-[~]
└─$ gobuster dir -u http://192.168.11.126:8080 -w ~/test_web.txt -t 5
===============================================================
Gobuster v3.8
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.11.126:8080
[+] Method:                  GET
[+] Threads:                 5
[+] Wordlist:                /home/chaimae/test_web.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.8
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
Progress: 16 / 16 (100.00%)
===============================================================
Finished
===============================================================
```

```
┌──(chaimae㉿kali)-[~]
└─$ nmap -p 8080 --script http-enum 192.168.11.126
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-03 04:49 EST
Nmap scan report for 192.168.11.126
Host is up (0.0013s latency).

PORT     STATE SERVICE
8080/tcp open  http-proxy
MAC Address: 08:00:27:A9:53:9F (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 23.43 seconds

┌──(chaimae㉿kali)-[~]
```

hydra -l chaimae -P ~/passwords.txt ssh://192.168.11.126 -t 4 -V



```
[RESPONDER] 📥 Décision reçue :
            Sévérité : high
            Catégorie : bruteforce
            Action : block_ip
            Raison : SSH brute-force detected (18 attempts)

[RESPONDER] 📥 Décision reçue :
            Sévérité : high
            Catégorie : bruteforce
            Action : block_ip
            Raison : SSH brute-force detected (19 attempts)

[RESPONDER] 📥 Décision reçue :
            Sévérité : high
            Catégorie : bruteforce
            Action : block_ip
            Raison : SSH brute-force detected (20 attempts)
```

```
            Action : alert
            Raison : Failed SSH connection attempt from a known IP address.
[RESPONDER] 🔔 Alerte Envoyée/Simulée: None (scan)
[RESPONDER] ✅ Log écrit dans /tmp/alerts.log
127.0.0.1 - - [03/Dec/2025 14:27:11] "POST /respond HTTP/1.1" 200 -
```

```
            Raison : Seuil SSH dépassé (8/5). Nécessite confirmation IA.
[RESPONDER] 🔔 Alerte Envoyée/Simulée: 192.168.11.140 (BRUTEFORCE_ATTEMPT)
[RESPONDER] ✅ Log écrit dans /tmp/alerts.log
127.0.0.1 - - [04/Dec/2025 21:32:29] "POST /respond HTTP/1.1" 200 -

[RESPONDER] 📥 Décision reçue :
            Sévérité : high
            Catégorie : BRUTEFORCE_ATTEMPT
            Action : alert
            Raison : Seuil SSH dépassé (9/5). Nécessite confirmation IA.
[RESPONDER] 🔔 Alerte Envoyée/Simulée: 192.168.11.140 (BRUTEFORCE_ATTEMPT)
[RESPONDER] ✅ Log écrit dans /tmp/alerts.log
127.0.0.1 - - [04/Dec/2025 21:32:32] "POST /respond HTTP/1.1" 200 -

[RESPONDER] 📥 Décision reçue :
            Sévérité : high
            Catégorie : BRUTEFORCE_ATTEMPT
            Action : alert
            Raison : Seuil SSH dépassé (10/5). Nécessite confirmation IA.
[RESPONDER] 🔔 Alerte Envoyée/Simulée: 192.168.11.140 (BRUTEFORCE_ATTEMPT)
[RESPONDER] ✅ Log écrit dans /tmp/alerts.log
127.0.0.1 - - [04/Dec/2025 21:32:33] "POST /respond HTTP/1.1" 200 -
```

🔴 ATTACK PHASE

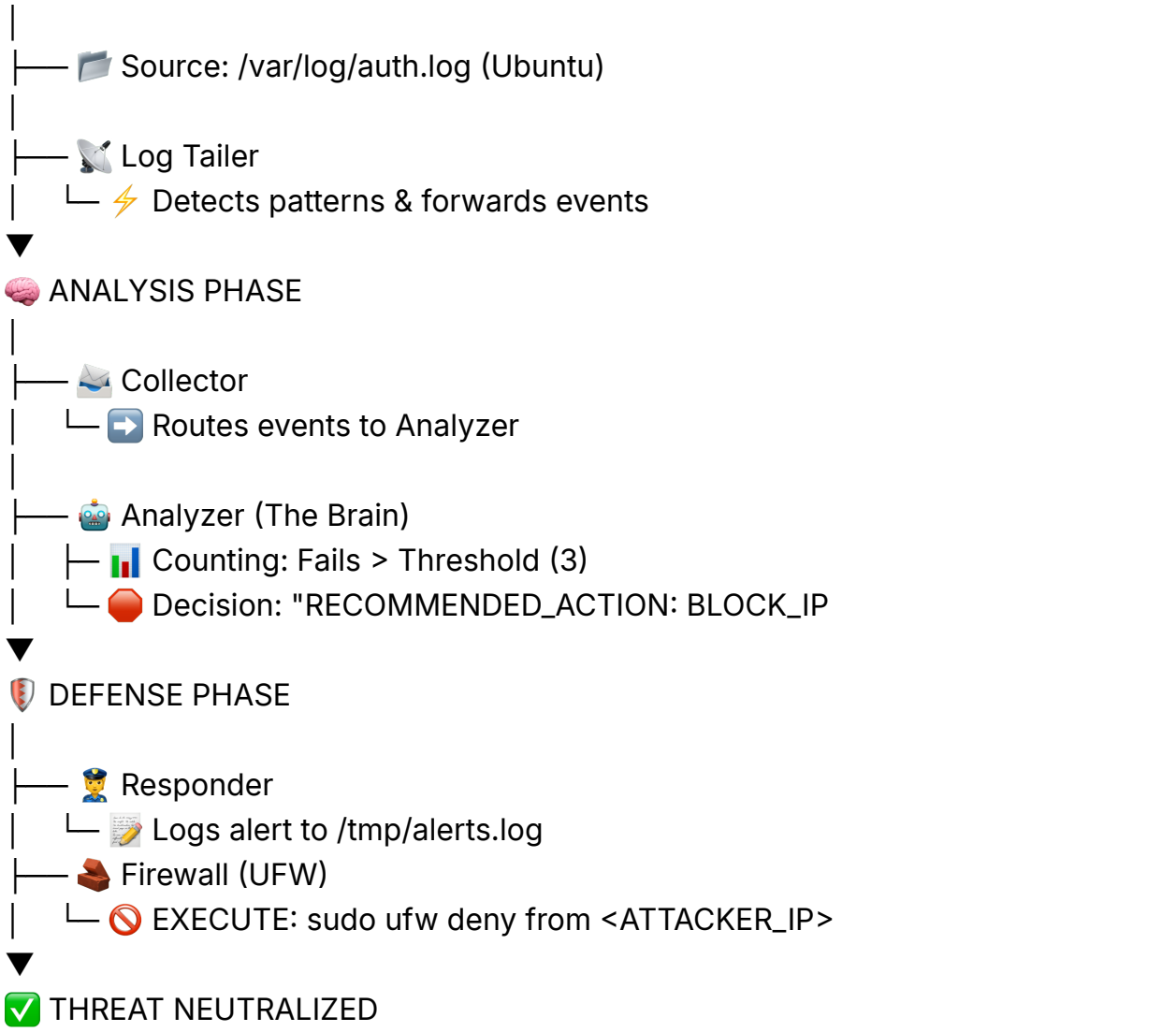
│

├── 💻 Attacker: Kali Linux (Hydra)

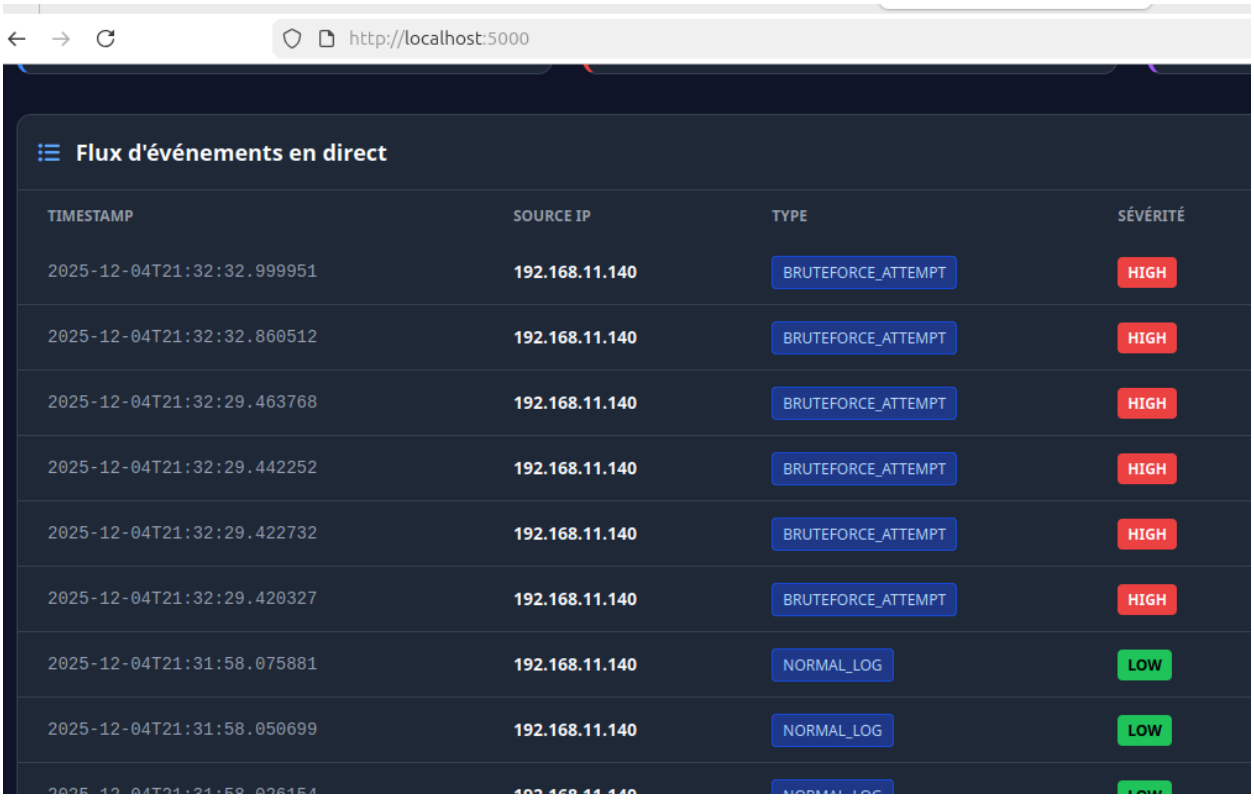
│ └── 🔥 Action: 10+ failed SSH login attempts

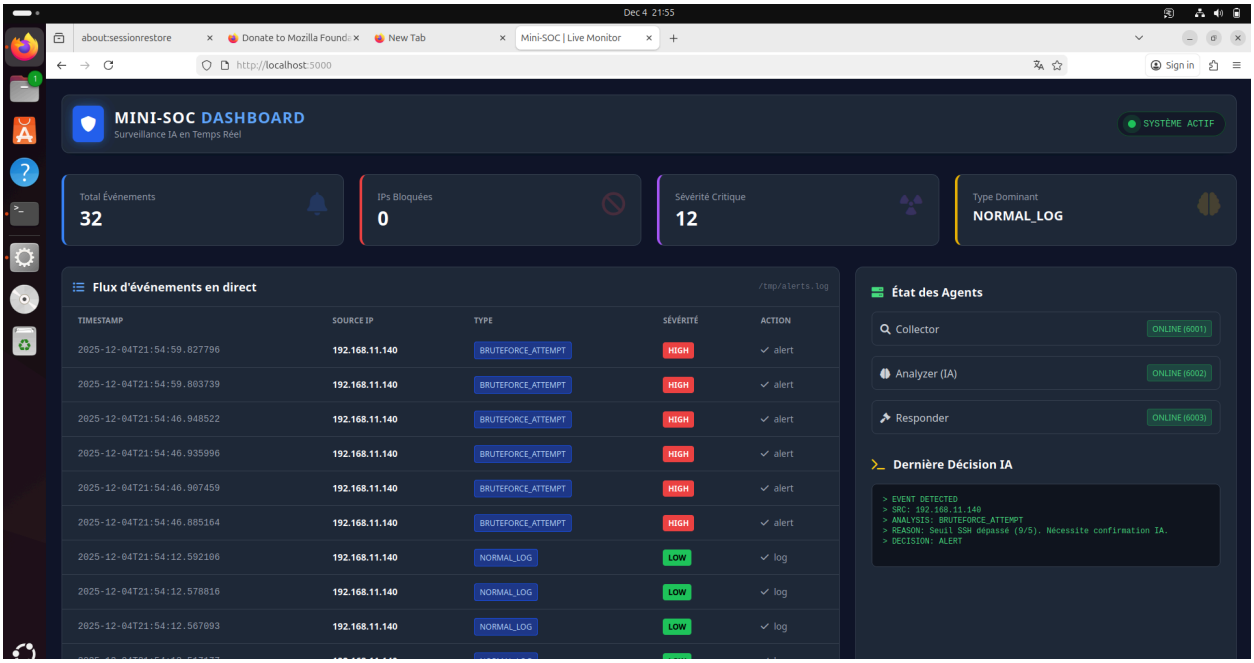
▼

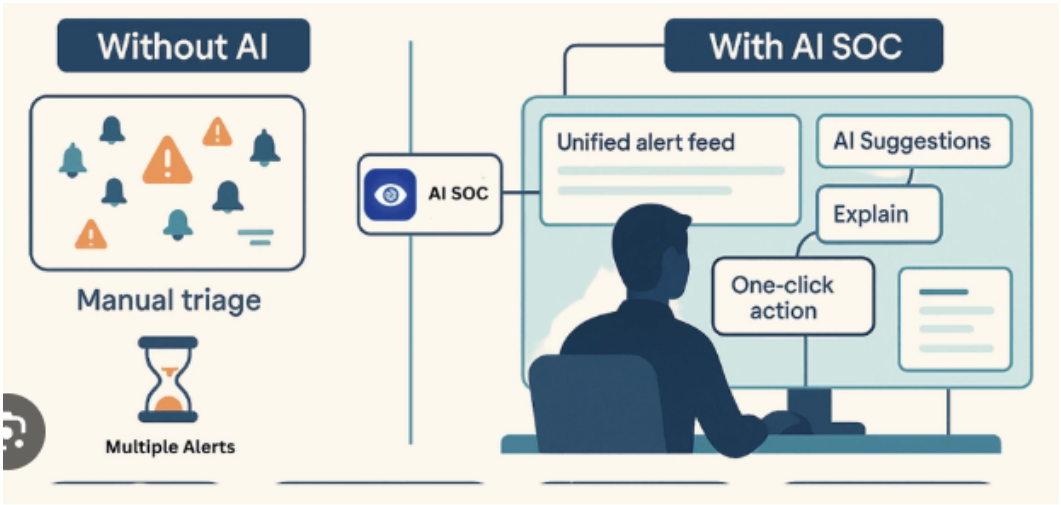
👁️ DETECTION PHASE


📂 Source: /var/log/auth.log (Ubuntu)
📡 Log Tailer
⚡ Detects patterns & forwards events
🧠 ANALYSIS PHASE
📩 Collector
➡️ Routes events to Analyzer
🤖 Analyzer (The Brain)
📊 Counting: Fails > Threshold (3)
🛑 Decision: "RECOMMENDED_ACTION: BLOCK_IP
🛡️ DEFENSE PHASE
👮 Responder
📝 Logs alert to /tmp/alerts.log
🧱 Firewall (UFW)
🚫 EXECUTE: sudo ufw deny from <ATTACKER_IP>
✅ THREAT NEUTRALIZED
MINI-SOC DASHBOARD
Surveillance IA en Temps Réel
SYSTÈME ACTIF
Total Événements
22
IPs Bloquées
0
Sévérité Critique
6
Type Dominant
NORMAL_LOG
Flux d'événements en direct
/tmp/alerts.log
TIMESTAMP
SOURCE IP
TYPE
SÉVÉRITÉ
ACTION
2025-12-04T21:32:32.999951
192.168.11.140
BRUTEFORCE_ATTEMPT
HIGH
unknown
2025-12-04T21:32:32.860512
192.168.11.140
BRUTEFORCE_ATTEMPT
HIGH
unknown
2025-12-04T21:32:29.463768
192.168.11.140
BRUTEFORCE_ATTEMPT
HIGH
unknown
2025-12-04T21:32:29.442252
192.168.11.140
BRUTEFORCE_ATTEMPT
HIGH
unknown
2025-12-04T21:32:29.422732
192.168.11.140
BRUTEFORCE_ATTEMPT
HIGH
unknown
2025-12-04T21:32:29.420327
192.168.11.140
BRUTEFORCE_ATTEMPT
HIGH
unknown
2025-12-04T21:31:58.075881
192.168.11.140
NORMAL_LOG
LOW
unknown
2025-12-04T21:31:58.050699
192.168.11.140
NORMAL_LOG
LOW
unknown
2025-12-04T21:31:58.026154
192.168.11.140
NORMAL_LOG
LOW
unknown
État des Agents
Collector
ONLINE (6001)
Analyzer (IA)
ONLINE (6002)
Responder
ONLINE (6003)
Dernière Décision IA
> EVENT DETECTED
> SRC: 192.168.11.140
> ANALYSIS: BRUTEFORCE_ATTEMPT
> REASON: N/A
> DECISION: UNKNOWN


État des Agents
Collector
ONLINE (6001)
Analyzer (IA)
ONLINE (6002)
Responder
ONLINE (6003)
Dernière Décision IA
> EVENT DETECTED
> SRC: 192.168.11.140
> ANALYSIS: BRUTEFORCE_ATTEMPT
> REASON: N/A


http://localhost:5000
Flux d'événements en direct
TIMESTAMP
SOURCE IP
TYPE
SÉVÉRITÉ
2025-12-04T21:32:32.999951
192.168.11.140
BRUTEFORCE_ATTEMPT
HIGH
2025-12-04T21:32:32.860512
192.168.11.140
BRUTEFORCE_ATTEMPT
HIGH
2025-12-04T21:32:29.463768
192.168.11.140
BRUTEFORCE_ATTEMPT
HIGH
2025-12-04T21:32:29.442252
192.168.11.140
BRUTEFORCE_ATTEMPT
HIGH
2025-12-04T21:32:29.422732
192.168.11.140
BRUTEFORCE_ATTEMPT
HIGH
2025-12-04T21:32:29.420327
192.168.11.140
BRUTEFORCE_ATTEMPT
HIGH
2025-12-04T21:31:58.075881
192.168.11.140
NORMAL_LOG
LOW
2025-12-04T21:31:58.050699
192.168.11.140
NORMAL_LOG
LOW





Conclusion

This project successfully demonstrates the viability of building an autonomous, privacy-centric Security Operations Center (SOC) using local resources. By integrating a modular Python architecture with a local Large Language Model (Mistral 7B via LM Studio), we achieved a functional defense loop capable of detecting, analyzing, and responding to cyber threats in real-time without relying on cloud-based SIEMs or external APIs.

Key Achievements:

- **Operational Autonomy:** The system operates independently, monitoring logs, classifying threats, and executing firewall blocks (`UFW`) without human intervention.
- **Privacy & Cost Efficiency:** Running the LLM locally ensures zero data leakage and eliminates recurring API costs, a critical advantage for sensitive environments.
- **Resilience via Hybrid Analysis:** The implementation of a fallback mechanism (heuristics) proved essential. When the LLM faced latency or timeouts during high-volume attacks (like Hydra brute-force), the system seamlessly switched to rule-based detection, ensuring continuous protection.
- **Real-Time Visualization:** The custom Dashboard provides immediate situational awareness, successfully displaying live threats and automated responses.