Skip to content

Chat history

ChatGPT
ChatGPT

Explore GPTs

Library
Today

- Security Analyst Project Ideas

Upgrade plan
More access to the best models

You said:

Security Analyst: This role involves monitoring and defending against cyber threats. In the UK, the average annual salary is approximately £45,000. In the US, Cyberseek.org indicates an average salary of $100,000 for a "Cyber Crime Analyst" which is listed as having common job titles including "Security Analyst" and "Information Security Analyst". However, a Reddit discussion suggests that true entry-level SOC analyst positions in the US might range from $50,000 - $7

This is my role I picked and I need a 3 project ideas that I can complete in 2 weeks. it shoudol be simple and also kinda

complex but I should be able to complete in 2 weeks and I am not form technical bacjround
ChatGPT said:
Got it! Since you're aiming for Security Analyst projects that are **achievable in 2 weeks**, **not super technical**, but still meaningful enough to show understanding of the role, here are 3 solid project ideas for you:

✅ 1. Simulate a SOC (Security Operations Center) with Free Tools

Goal: Show how a SOC team monitors and responds to threats.

What You'll Do:

- Use a tool like Security Onion or Splunk Free.
- Simulate network traffic using Wireshark or simple attack tools like Metasploit (light usage).
- Detect basic anomalies (unauthorized access attempts, suspicious logins).
- Document how alerts are generated and how you'd respond.

Why it's good: Gives you hands-on with tools real analysts use, without needing deep coding. You'll learn about logs, alerts, and response steps.

✅ 2. Create a Cyber Threat Report (like you're in a SOC team)

Goal: Research and document a current cyber threat and how an organization could respond.

What You'll Do:

- Pick a real-world threat (e.g., ransomware, phishing campaign, or a recent data breach).

- Use OSINT (Open-Source Intelligence) tools like VirusTotal, Shodan, or HaveIBeenPwned.
- Write a 3-5 page threat intelligence report: what happened, indicators of compromise (IoCs), and mitigation steps.
- Bonus: Format it like a real incident report (you can find templates online).

Why it's good: No coding needed, but it's still analytical. Shows you understand how to track threats and communicate them.

✅ 3. Build a Home Lab for Basic Threat Detection

Goal: Set up a virtual lab to monitor and identify suspicious activity on your own system.

What You'll Do:

- Set up a simple Windows + Linux VM (use VirtualBox).
- Install and configure **Sysmon** and **Windows Event Viewer** to log system activity.
- Perform some normal + suspicious actions (file downloads, privilege escalation simulations).
- Analyze logs and write a report on what you detected.

Why it's good: Teaches you the basics of log analysis and how attackers leave traces — key Security Analyst skills. No deep programming needed.

Would you like help picking which one fits you best based on your current skills or time? Or help setting one of these up step by step?

You said:

✅ 1. Simulate a SOC (Security Operations Center) with Free Tools
Goal: Show how a SOC team monitors and responds to threats.

What You’ll Do:

Use a tool like Security Onion or Splunk Free.
Simulate network traffic using Wireshark or simple attack tools like Metasploit (light usage).
Detect basic anomalies (unauthorized access attempts, suspicious logins).
Document how alerts are generated and how you'd respond.
Why it’s good: Gives you hands-on with tools real analysts use, without needing deep coding. You’ll learn about logs, alerts, and response steps.

I am picking this and I need detailed 2 weeks plan for this and detailed project plan what to do and all
ChatGPT said:

Updated saved memory
Awesome choice! Simulating a SOC is a fantastic hands-on project that gives you real-world insights and is perfect for someone starting out. Here’s your **detailed 2-week plan** broken into daily tasks, along with a **project overview** and what to deliver at the end.

🧩 Project Title:

Simulating a Security Operations Center Using Free Tools

🎯 Project Goal:

To simulate how a SOC team detects and responds to cyber threats using free tools (e.g., Splunk Free or Security Onion), basic traffic simulation, and alert documentation — all without needing advanced technical skills.

📆 2-Week Plan (Day-by-Day Breakdown)

🔷 Week 1: Setup, Learning Tools, Simulating Traffic

Day 1-2: Setup Environment

- Install **VirtualBox** or **VMware** on your machine.
- Install one or more of the following inside a VM:
    - **Splunk Free**
    - OR **Security Onion**
    - Optional: Set up a second VM with **Kali Linux** for light simulation.
- Install **Wireshark** on host or in VM for packet analysis.

📝 Tip: Splunk is easier if you're new; Security Onion is more complex but preloaded with SOC tools.

Day 3: Understand the Tool

- Watch a Splunk beginner tutorial (YouTube has tons!)
- Learn:
    - How to collect logs (Windows Event Logs, Syslog, etc.)
    - How to search data in Splunk (use `index=*` to start exploring)
- Load test logs into Splunk (you can use sample datasets)

Day 4-5: Simulate Network Traffic

- Use **Wireshark** to observe live traffic on your system (browsing, downloads, etc.).

- Install **Sysmon** (Windows tool) to generate detailed logs.
  - Simulate normal activities: browsing, logins, opening files.
  - Simulate suspicious behavior:
    - Create a fake admin user.
    - Run a simple PowerShell command to mimic malware activity.
- Let Splunk or Security Onion collect and analyze the logs.

Day 6-7: Basic Detection

- Identify:
  - Unusual login times.
  - Privilege escalation.
  - Suspicious IP connections.
- Use search queries like:

```
index=main EventCode=4625
```

- `index=main "powershell"`

- Begin taking notes on the alerts or anomalies you detect.

🔷 Week 2: Alert Analysis, Documentation, Final Report

Day 8-9: SOC Workflow Simulation

- Choose 1–2 real-world attack scenarios (e.g., brute-force login or phishing).
- Simulate the behavior lightly or find sample logs.
- Walk through the process:
  - Detection → Alert → Triage → Response

🛠️ Example: Failed login attempts → detected in Splunk → SOC analyst investigates → escalates if needed.

Day 10-11: Document Alerts and Responses

- For each detected anomaly or simulated event:
  - Describe the alert.
  - Add screenshots of logs/queries.
  - Write how you'd respond:
    - Block IP?
    - Disable account?
    - Monitor system?
- Use an **Incident Report Template** (I can provide one).

Day 12: Wrap-Up Analysis

- Review what you built and learned:
  - How effective was detection?
  - What worked well or didn't?
- Compare with what a real SOC would do.
- Optional: Create a short slide deck.

Day 13-14: Final Report / Presentation

- Write your **Final Report** (PDF or Google Doc):
  - Introduction (your goal)
  - Tool setup screenshots
  - Simulated events
  - Logs, alerts, actions
  - Conclusion + lessons learned
- Optional: Record a short 2–3 min video demo or voiceover slide deck.

📦 What You'll Deliver at the End

- ✅ Final Report (PDF/Doc)
- ✅ Screenshots of Splunk/Security Onion dashboards
- ✅ At least 2–3 simulated alerts with your investigation notes
- ✅ Optional: short PowerPoint-style summary

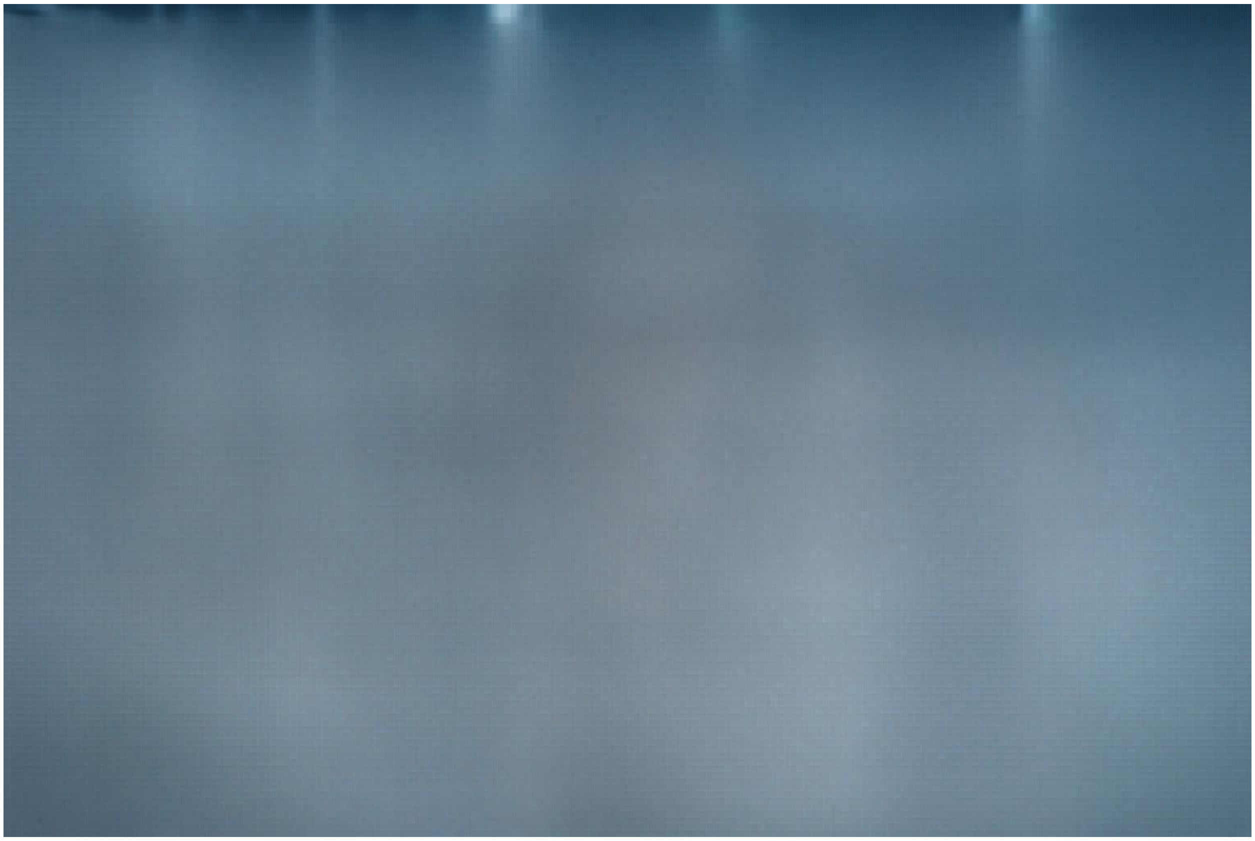
If you want, I can generate a **template report document**, example alerts, or sample search queries to help you move faster. Want that?

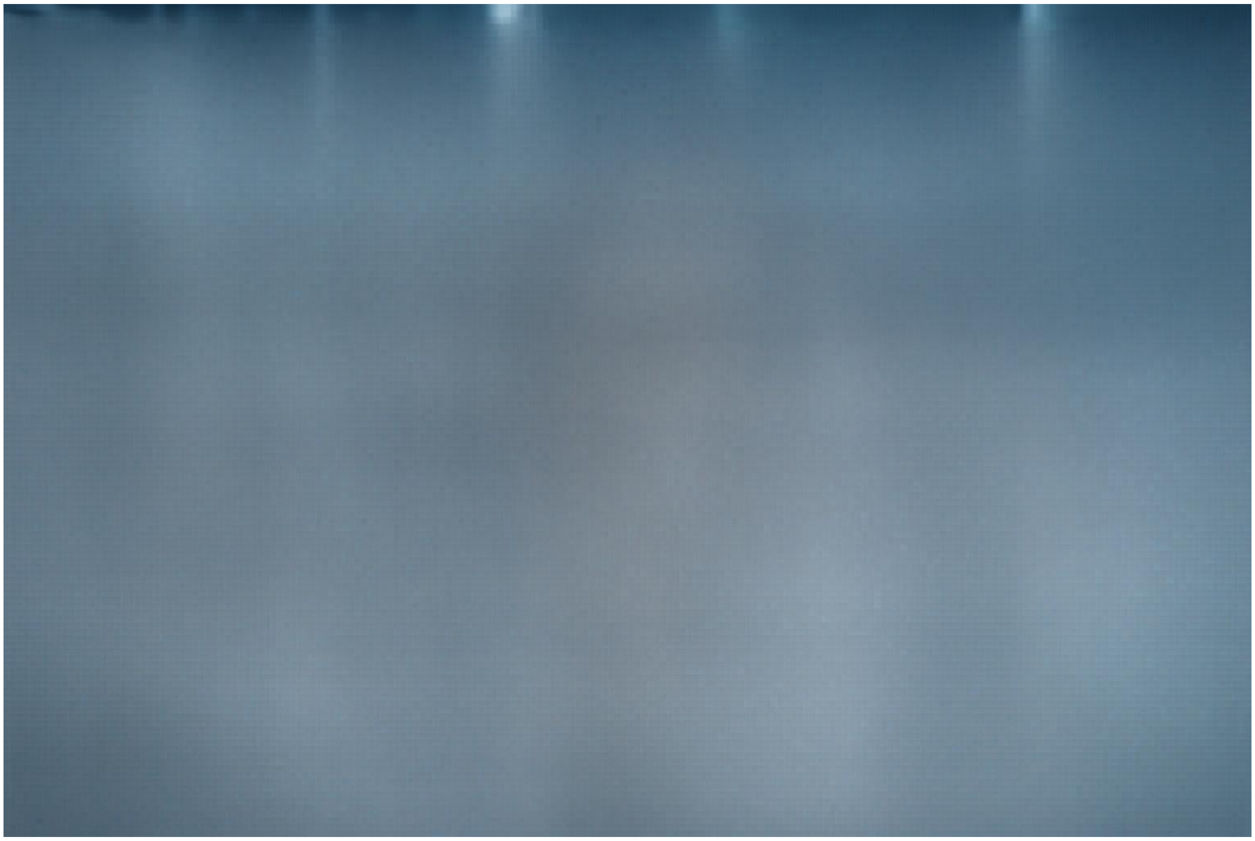
You said:

Create an image on how it loos like in the real life this project.

ChatGPT said:

Image created



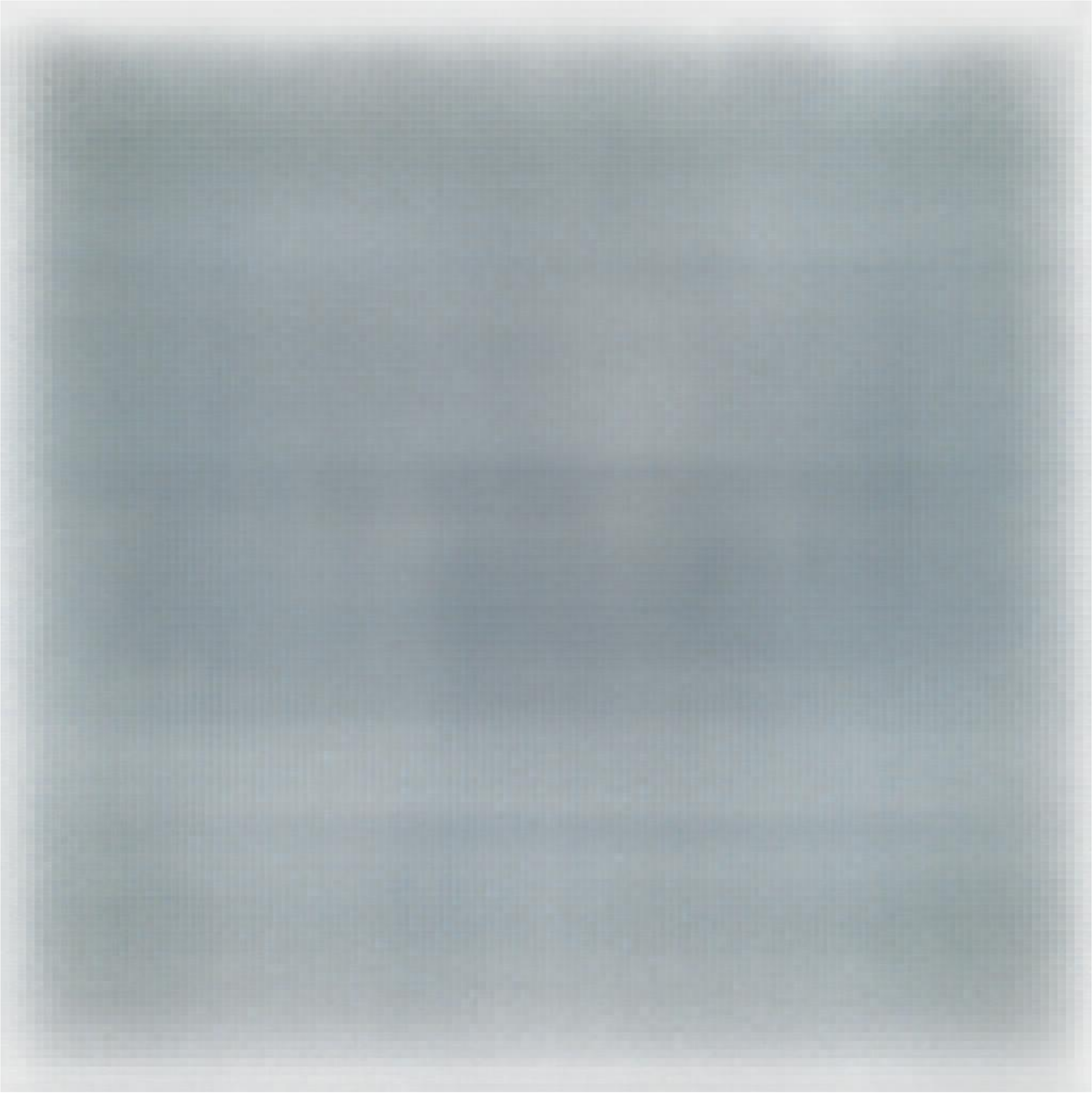


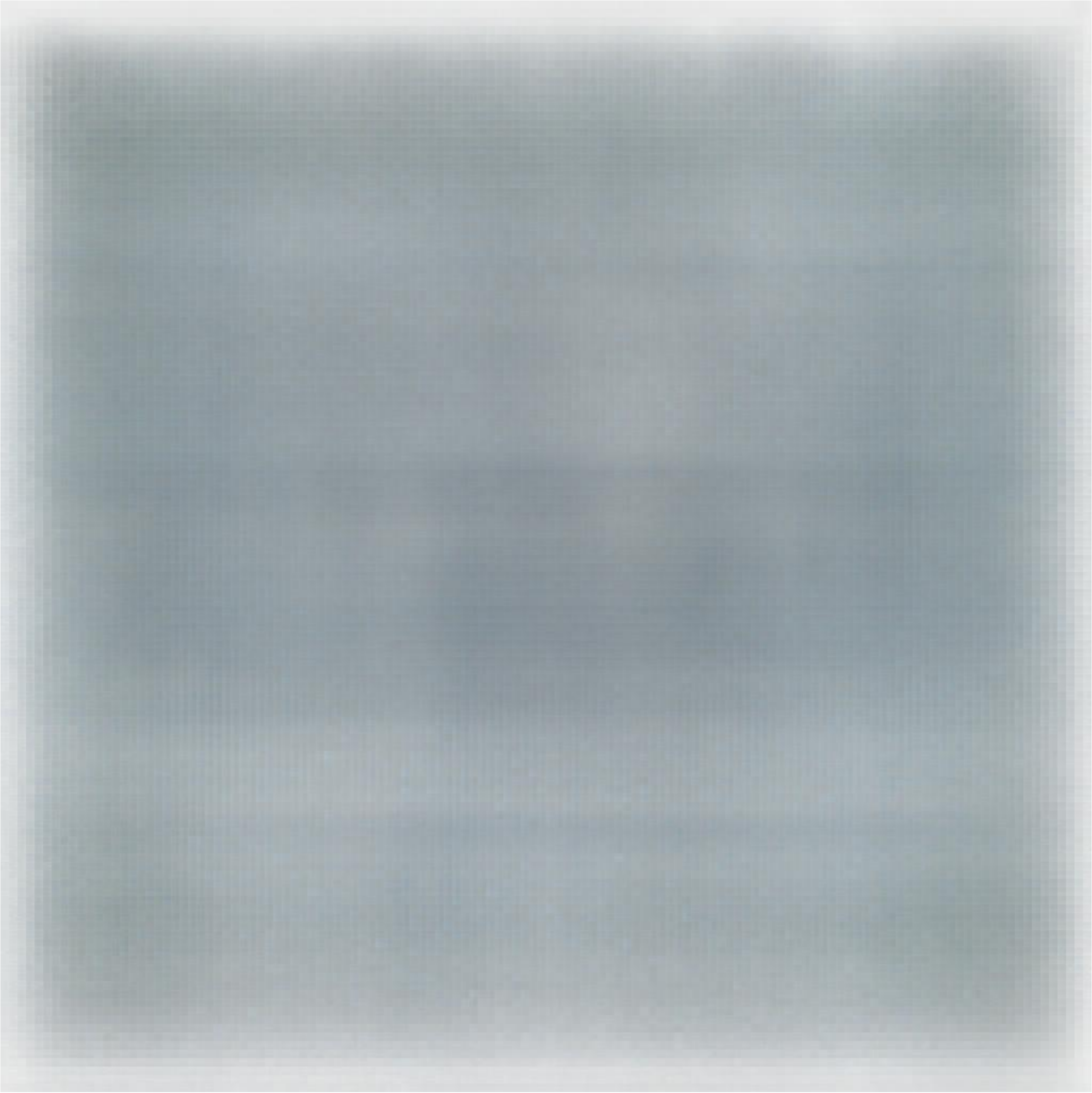
You said:

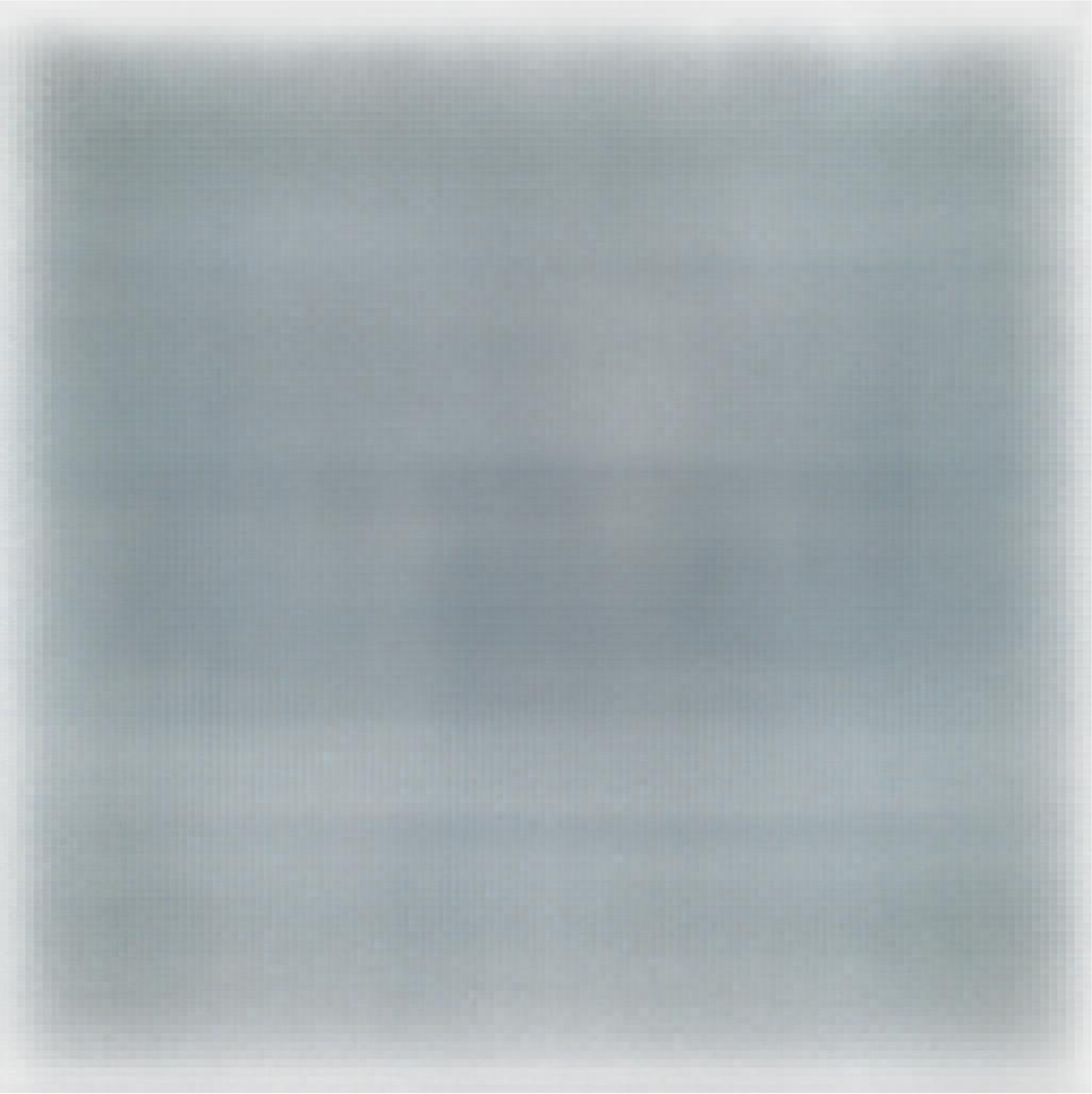
create an image of timeline as well

ChatGPT said:

Getting started







You said:

Gice me a code that helps SOC analysts

ChatGPT said:

Sure! Here's a simple but **powerful Python script** that's useful for SOC analysts — it parses and analyzes **Windows Event Logs** (from `.evtx` files), which are common in threat detection and forensic investigations.

🛠️ Python Script: Extract Suspicious Logins from Windows Event Logs

This script looks for **failed logins** (Event ID 4625) and **successful logins** (Event ID 4624), helping a SOC analyst identify brute force attacks, unauthorized access, or unusual logon behavior.

✅ Requirements:

- Python 3.x
- `evtx` and `xmltodict` libraries

Install with:

```
pip install python-evtx xmltodict
```

🧠 Code:

```
from Evtx.Evtx import Evtx
import xmltodict

def parse_event_log(file_path):
    with Evtx(file_path) as log:
        for record in log.records():
            data = xmltodict.parse(record.xml())
            try:
                system_data = data['Event']['System']
                event_id = int(system_data['EventID']['#text'])
                time_created = system_data['TimeCreated']['@SystemTime']
                computer = system_data['Computer']

                if event_id in [4624, 4625]:  # 4624 = successful login, 4625 = failed login
```

```
                    event_data = data['Event']['EventData']['Data']
                    username = ""
                    ip_address = ""

                    for entry in event_data:
                        if entry['@Name'] == 'TargetUserName':
                            username = entry['#text']
                        if entry['@Name'] == 'IpAddress':
                            ip_address = entry['#text']

                    print(f"[{time_created}] Event ID: {event_id} | User: {username} | IP: {ip_address} | Computer: {computer}")

            except Exception as e:
                continue

# Example usage:
parse_event_log("security.evtx")
```

📊 What It Does:

- Extracts log time, username, IP address, and event type (success/fail).
- Helps spot brute force attempts, unusual login times, or foreign IPs.

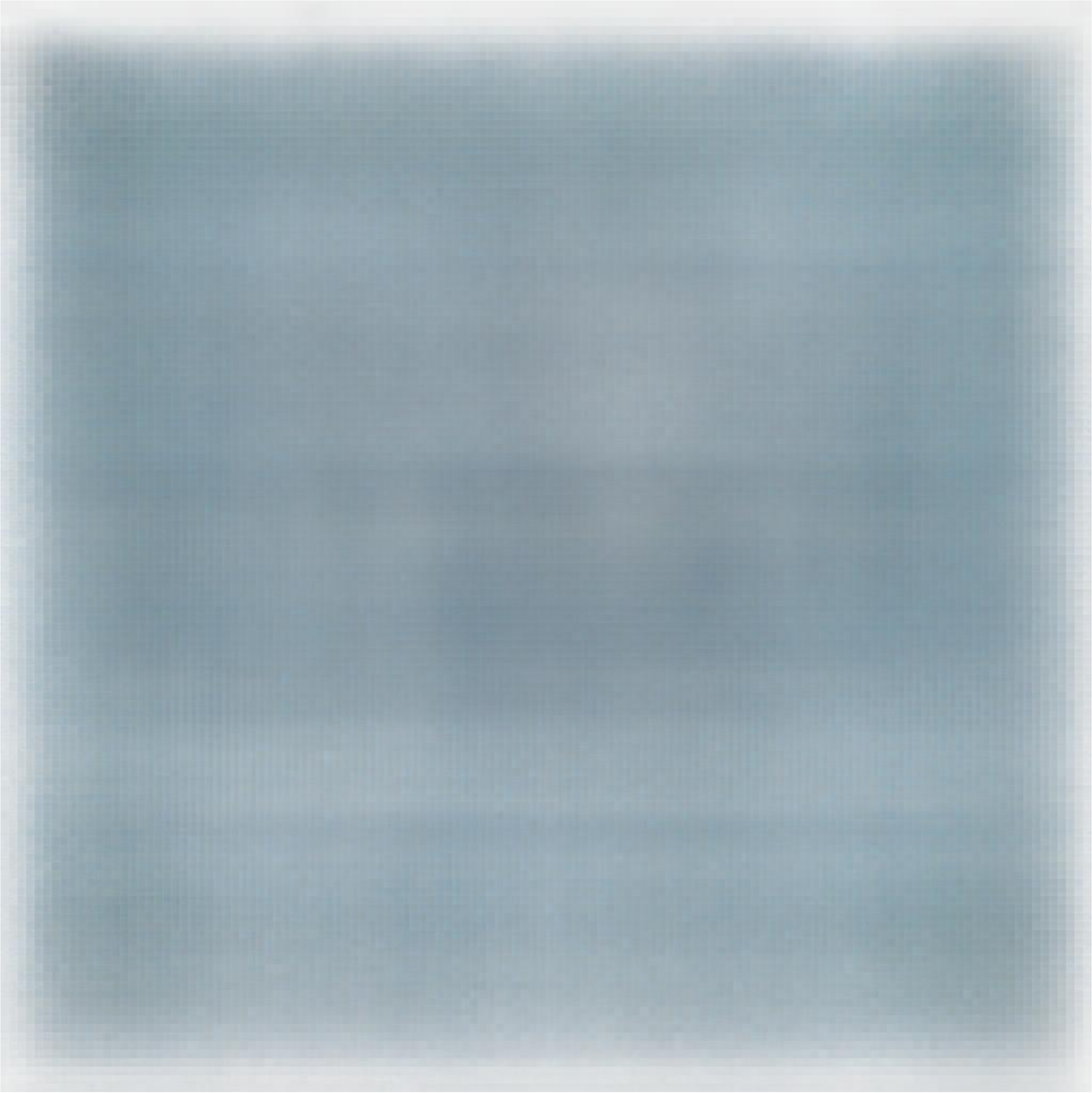
Want help customizing this for other events like **process creation (4688)** or **USB insertions (2003)**? Or want a version that saves results to a CSV file or dashboard?

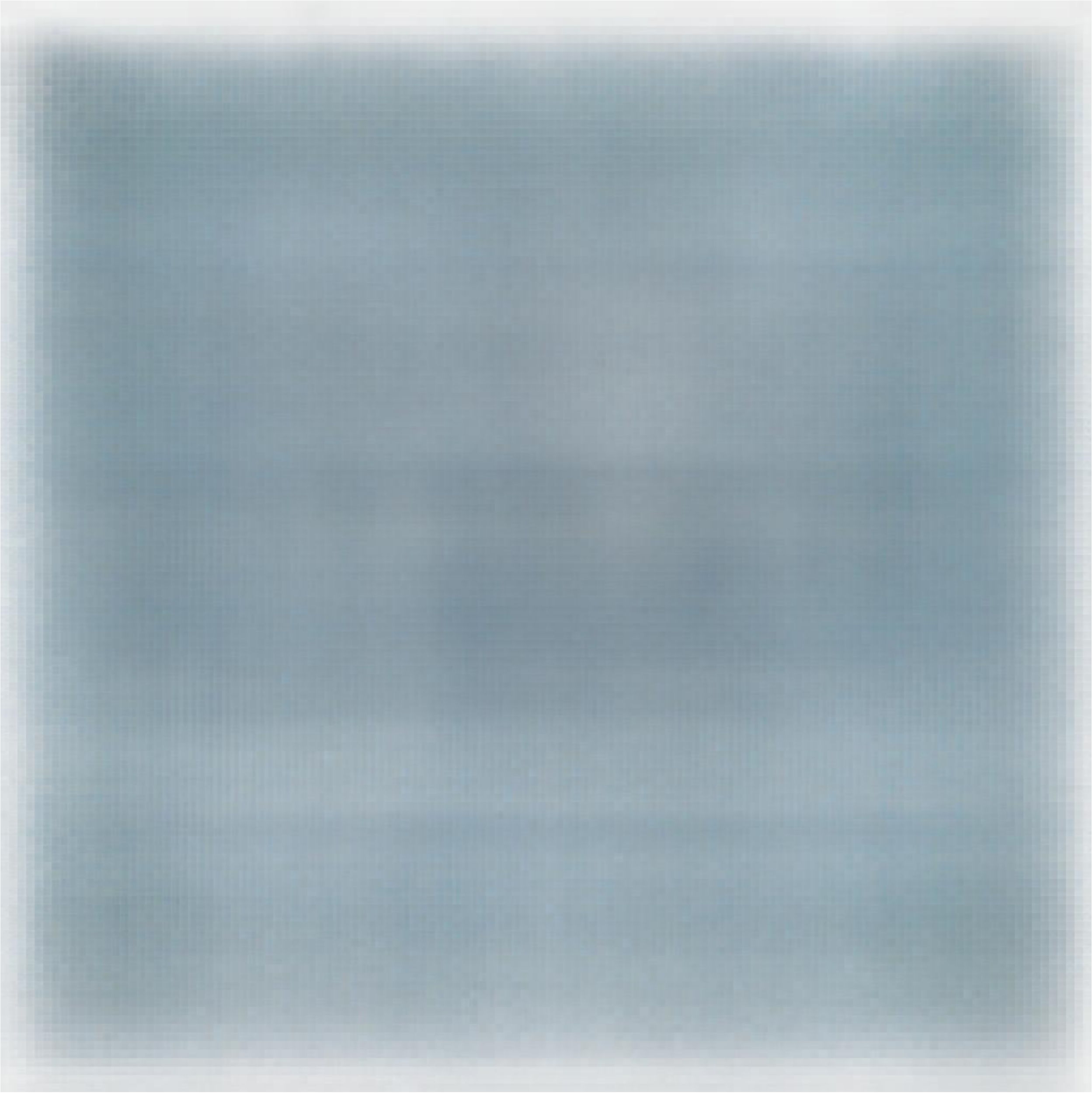
You said:

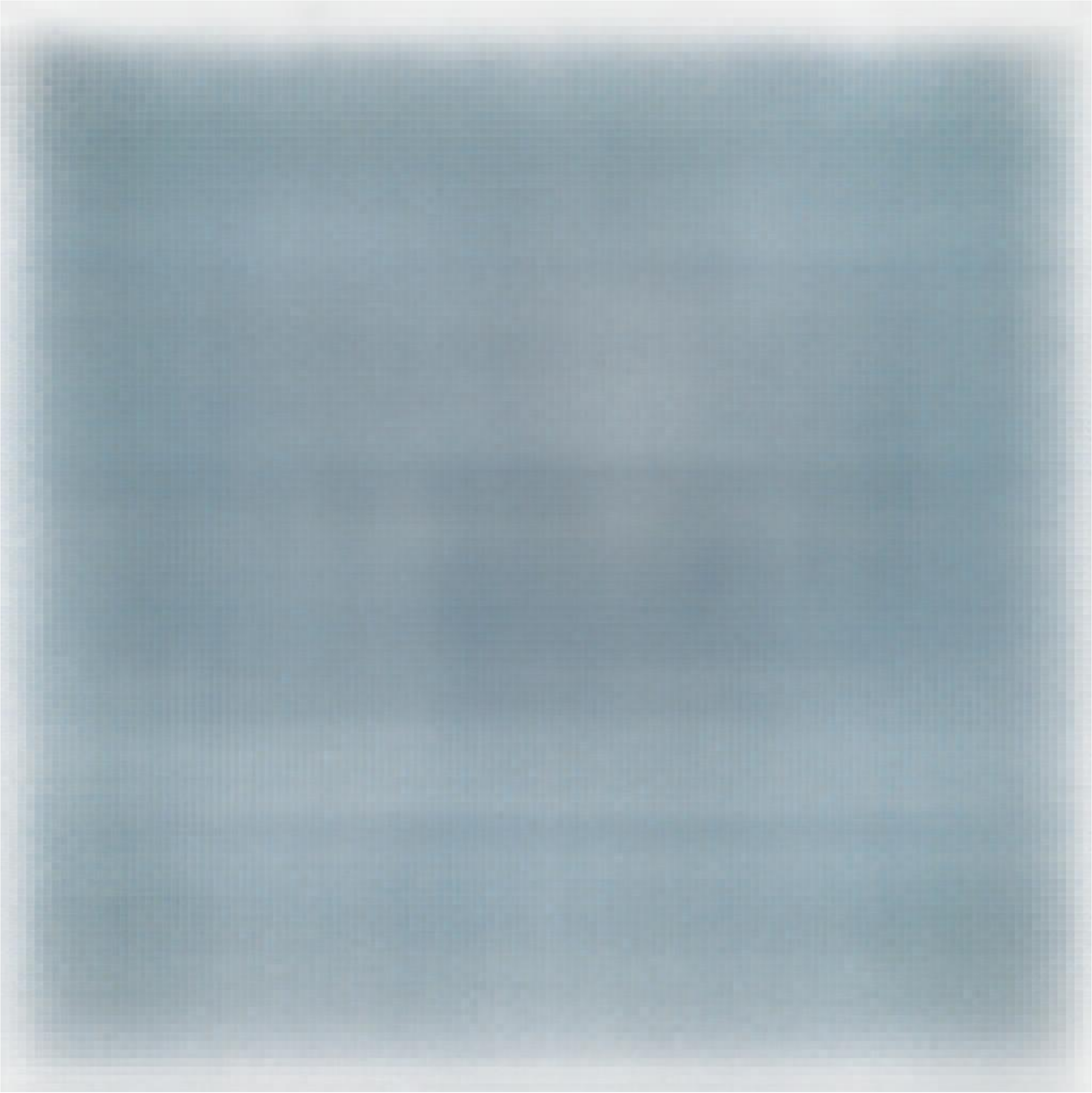
CREATE AN IMAGE OF TIME LINE

ChatGPT said:

Getting started







Search

Reason

ChatGPT can make mistakes. Check important info.

?

ChatGPT is still generating a response...