Cybernetic Showdown: Safeguarding Sensitive Information from Skynet

Author: Ryan McDonald j.ryan.mcdonald@gmail.com
Advisor: John Hally

Accepted: 12/7/2023

Abstract

The use of generative artificial intelligence has become extremely popular recently. This has presented a difficult problem for organizations that seek to embrace the use of the technology and the improved efficiency it provides, but it also introduces risk to the organization. There have been many high-profile reports of sensitive information being uploaded to these applications. As a result, many organizations have taken the approach of blocking all access to the technology. While this does stop users from uploading sensitive information, it also removes the ability to take advantage of the added efficiency the technology provides. This research looks for an alternative tool that can be used within an organization that also blocks users from uploading sensitive information.

1. Introduction

With the recent explosion of generative artificial intelligence (GenAI) tools many have seen the efficiency benefits of its use. The productivity benefits are obvious—it can perform research, write code, and take meeting notes. These types of tasks are usually manual and require significant time to complete, so automating most tasks can make employees much more efficient. While these tools have many benefits, some organizations have also started to realize that using these services can also be very risky. As a result, organizations have started to ban GenAI usage from corporate devices.

A recent Wall Street Journal article discussed how Apple, JPMorgan, and Verizon had all barred using these types of technologies (Tilley, 2023). When users ask questions or input data into these prompts the information they enter is captured by the system. Developers and engineers review and use these responses to improve the product. Companies are concerned that users could unintentionally share proprietary or confidential information. They are also worried about breaking regulatory compliance if a user leaks protected information (Fern, 2023).

While blocking all access is one way to mitigate this risk, it does so at the expense of losing all the benefits this transformative technology provides (Derose, 2023). A much better approach would be to allow use but filter what users are allowed to enter in the prompt. This allows users to utilize the benefits while reducing the risk. There are commercially available security tools that provide this capability, but they are very expensive and may not provide the ability to customize the blocking or alerting within their platform. The purpose of this research paper is to demonstrate how organizations can use open-source or free security tools to provide access to these business-enabling technologies but do so in a manner that does not also introduce significant risk.

2. Research Method

When analyzing the problem, the method was first to understand the problem and how users interact with common GenAI sites such as ChatGPT. Next, research was

performed to know if there were any tools or capabilities already on the market that could solve this problem. If so, what were their limitations and costs? After understanding the problem and solutions on the market, the commercial solutions could then be compared to open-source or free tools to see if any could provide the same or better level of protection. After choosing a solution and various configurations, the use cases were tested to see if the hypothesis was correct and to record any findings.

2.1. Understand the Problem

OpenAI's ChatGPT was chosen for this research since it is very well-known. Research began by interacting with the website and doing research on the network traffic profile to understand the protocols used and how the website functioned. There was a great breakdown on the keysight.com blog of ChatGPT's traffic (Sahu, 2023). Some of the important findings are the website traffic is SSL/TLS encrypted, QUIC version 1 is used to chat with the chatbot, and the input or question is sent through a POST request to the conversation API of the chat.openai.com server.

Another item that needed to be understood was what were common risky behaviors that an organization would want to monitor. The most common were related to inputting financial data and PII, uploading documents (spreadsheets, word documents, PDFs, etc.) with sensitive information, and uploading source code of proprietary applications. This will be important in designing the protections or monitoring use cases.

2.2. Analyzing Solutions

After understanding the problem and defining the risky behaviors, possible solutions began to come into focus. The current commercial tools were evaluated along with their pros and cons. Then, various free and open-source tools were compared to understand which may best suit the current requirements and use cases. After looking at the options, both the network and host solutions came with their respective pros and cons. Here are some of the requirements for this testing to be successful:

1. Visibility into encrypted network traffic (Network)
2. IDS/IPS rules of the defined risky behavior (Network)
3. Agent or browser extension to filter content before leaving the host (Host)

4. Forward web traffic logs to SIEM for monitoring alerts (Network or Host)

The main challenge from the network perspective is SSL/TLS decryption. It is difficult to do, and adding the QUIC protocol makes it more complicated. From the host's perspective, the biggest challenge is coding ability and the time it takes.

2.3. Experiments and Testing

Based on the requirements, multiple tools have the capability needed. Below is a table displaying the requirements and associated tools needed:

Capability	Potential Tool
SSL/TLS Inspection	Squid Proxy, Wireshark, Fiddler Proxy, Burpsuite, PolarProxy
IDS/IPS	Suricata, SNORT, Zeek
SIEM	Elastic Logstash Kibana (ELK), Spunk (Community Edition)

Figure 1. Tool Capability Matrix

3. Findings and Discussion

The problem trying to be solved is to allow access to generative AI websites, such as ChatGPT while reducing the risk that a user could enter sensitive information. Protections can be implemented at the host or network level. It is also possible to forward logs from the host or network level to a SIEM to create alerts that detect when sensitive information has been inputted.

3.1. Network Traffic Profile

The first step in beginning the testing phase is to understand what the network traffic profile looks like. This will inform which tools may be better suited for a particular use case. It is also important to know how to create alert or blocking rules based on the traffic. When creating alerts or blocking rules, it is better to be as granular as possible to reduce false positives and ensure legitimate business traffic is not blocked.

A combination of Burpsuite and open-source research was used to understand the network traffic profile for chat.openai.com. Burpsuite is a great tool for this as it has a built-in proxy and browser that make it much easier to see HTTP and HTTPS traffic.

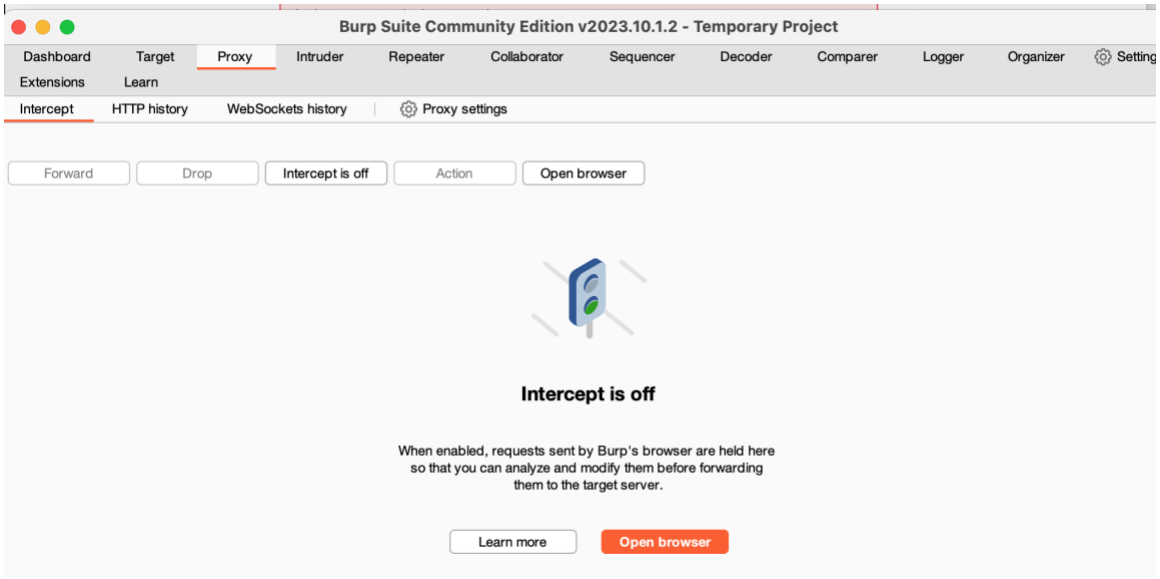


Figure 2. Burpsuite Proxy GUI

While Burpsuite is a commercial tool, there is a free version that can be used. The functionality is limited compared to the commercial version, but it can handle this type of task easily. It makes it easy to filter based on GET or POST requests, and also shows the IPs, full URI path, and many more details about the traffic.

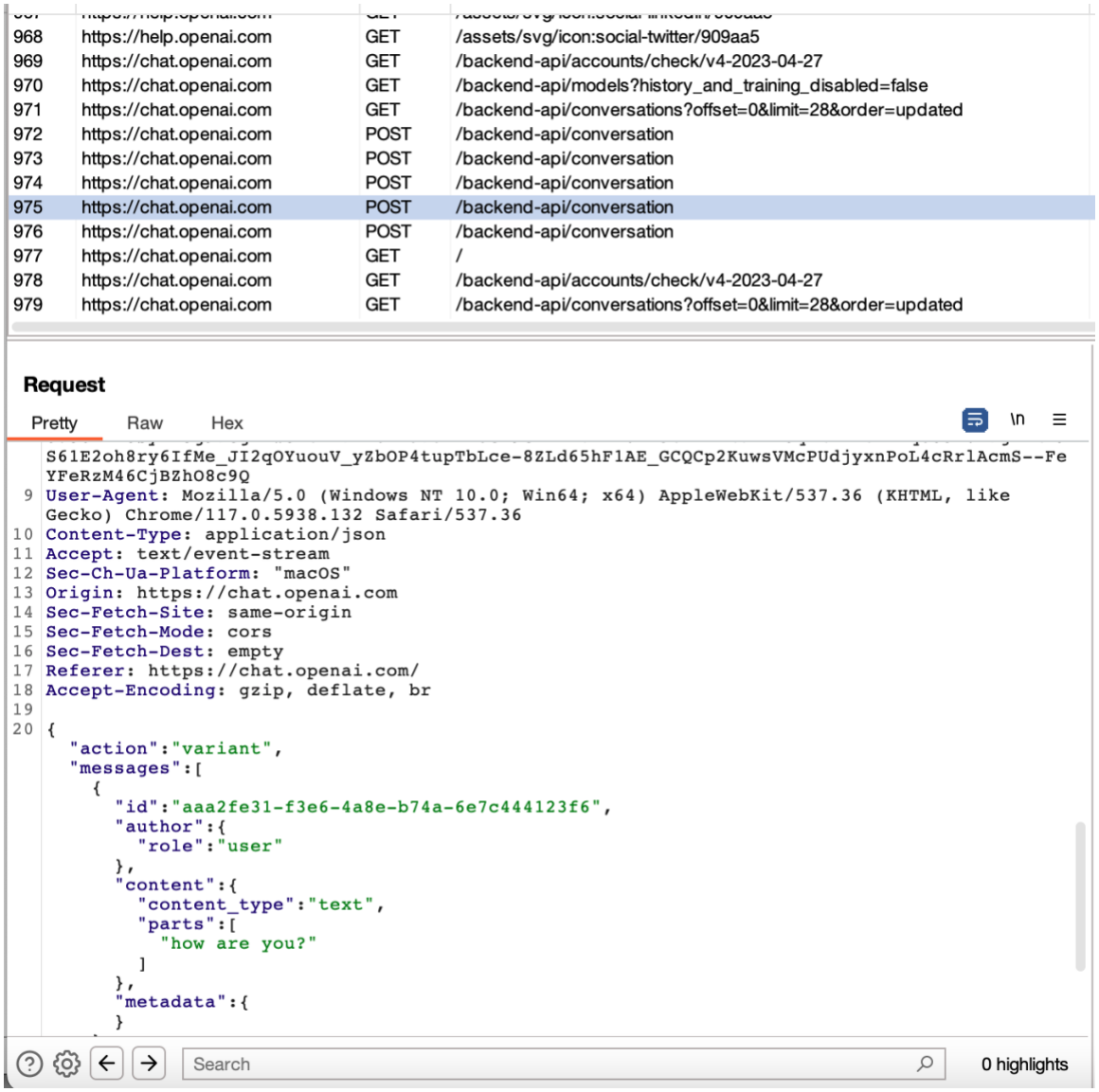


Figure 3. Burpsuite Proxy GUI with HTTPS Traffic

Within the body of the HTTPS packet, the actual text of the question submitted to chat.openai.com can be seen. It is important to take note of this field because it will be the focus for writing alerts or parsing within a SIEM for monitoring. See Figure 4, below.

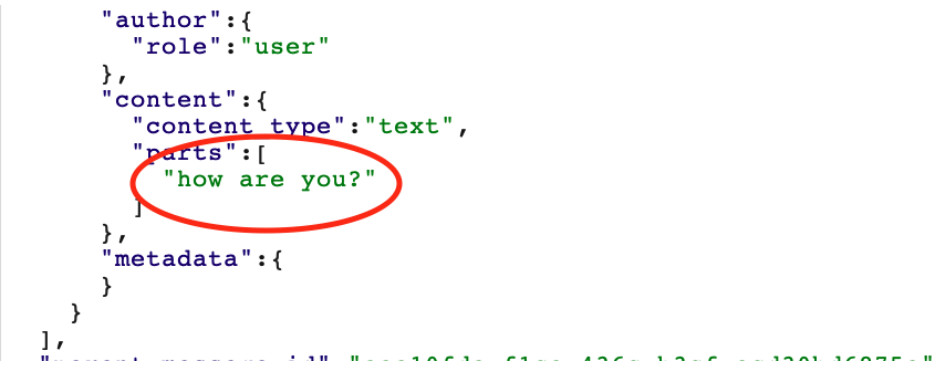


Figure 4. ChatGPT Prompt Text

3.2. Tools

After profiling the network traffic and understanding the telemetry, tools can now be evaluated to see which would work best. The full list of tools that were considered can be found in Figure 1: Tool Capability Matrix found in Section 2.3.

3.2.1. IPS/IDS

The IPS/IDS tools listed are all very capable, but there are some slight differences between the three. Zeek (formerly Bro) is a pure IDS and does not have IPS capabilities. Since it is much better to block this type of activity versus alerting, Snort or Suricata would be more beneficial in this scenario. Snort and Suricata are both very similar and the rule structure is as well. They can both serve as an IDS or IPS and have a large support community. The main difference between the two is Suricata was developed to make up for some shortcomings found in Snort. Most importantly, it is multi-threaded, and Snort is single-threaded. Due to these reasons, Suricata was chosen for this research.

3.2.2. Proxy

There were multiple proxies to choose from and multiple used throughout the project. Burpsuite's built-in proxy was used in profiling the network traffic as well as to troubleshoot issues when they arose. Squid proxy and Fiddler proxy were used as well for SSL inspection. Seeing the encrypted traffic was one of the most challenging parts of this project, so all three were used in some capacity. In the end, PolarProxy worked best for the identified use cases and was able to provide the SSL/TLS inspection needed for the project.

3.2.3. Browser Extension

Throughout the course of the network testing there were many challenges. Implementing a solution like this with open-source tools requires a high degree of technical expertise to execute in a way where the solution works as expected and does not affect normal business traffic. Also, due to the changing work environment of more and more employees working outside of the office, there are many shortcomings to implementing a pure network solution. When taking all these factors into consideration, having a host-based solution could be very beneficial to a lot of organizations. After researching it was discovered that an open-source browser extension named LakeraAI had many of the core capabilities already built into the product. It can also be customized to fit other use cases if the default ones are insufficient. The GitHub repository can be found here: https://github.com/lakeraai/chrome-extension and it can also be downloaded from the Chrome Store here: https://chrome.google.com/webstore/detail/lakera-chatgpt-data-leak/npdeilagbbimhnbbdjmagmedchnpjeid .

Some of the native capabilities are being able to block or obfuscate the following data types:

- Credit card numbers
- Anglophone names
- Email addresses
- Phone numbers
- US street addresses
- US social security numbers
- Secret keys

3.2.4. Other tools

Wireshark and Splunk (Community Edition) were also used in testing. Wireshark was used to capture and inspect the traffic, and Splunk as the SIEM for logging data and events. Wireshark is extremely useful because it can capture the traffic, filter out some of

the noise, and then allow the replay the traffic in Suricata to check if alerts are tuned correctly. There are some limitations to the Splunk Community Edition compared to the full commercial version. The main issue is limits on the amount of data one can put in per day (max 500 MB), but there are also some alerting capabilities that do not work in this version.

3.3. Defining Risky Behaviors

There are many risks associated with Generative AI technologies, but for the scope of this paper, at the focus will be on risks associated with user inputs. One type of risk concerns compliance, such as a user inadvertently inputting financial data, Personally Identifiable Information (PII), and medical data. Another risk is the inputting of intellectual property or company proprietary information. One example is the source code of a company-created application or details of a possible merger and acquisition. Leaking this type of information could be very costly.

One of the main advantages of using open-source tools is the ability to customize to an organization's specific requirements. Financial data and PII are straightforward use cases. Others that a company may want to consider could be a dirty word search of terms associated with highly sensitive data or programs. Another approach could be to block users from inputting specific document types, such as Excel or PDF documents. Finally, restricting code (Python, Java, etc.) from being inputted may be another way to safeguard company information. The important thing is that a company knows the data they are trying to protect and then builds use cases to protect that type of data.

3.4. Testing

There are benefits and drawbacks to going with the network or host-based protections. Network-based protections are beneficial in that they do not require coding. The solutions are comprehensive and provide a high degree of customization when implementing rules. Another benefit is there are a variety of solutions to choose from with the most common being Snort, Zeek, and Suricata. However, they still require a significant amount of technical expertise to deploy and maintain. It is not uncommon for network security solutions to introduce delays or block normal business traffic. Another

drawback of network-based solutions is that encryption is getting better and, thus, harder to perform SSL/TLS inspection, which can limit some of the effectiveness of the security controls. Even where SSL/TLS inspection is possible, it introduces privacy issues for the organization. This is why ensuring that network-based solutions are implemented thoughtfully and tested thoroughly is important.

Open-source host-based solutions are much harder to find and the functionality seems much more limited than the network solutions. These solutions can either exist as agents running on the host or as browser extensions. Another downside of open-source host-based solutions is they can require a significant amount of coding and endpoint expertise to customize existing solutions for specific use cases. However, there are benefits to using these types of solutions as well. One benefit of host-based solutions is it is possible to see some kinds of traffic before it is encrypted. It also can pop up customized warning messages for users to help with training around security. Finally, the main benefit to these types of protections is that they follow the user whether they are on the company network or not, which is very important in today's world of remote working.

3.4.1. Network-Based Testing

Network-based testing was approached from many angles due to the challenges provided by SSL/TLS inspection. The original plan was to implement Security Onion, which contains a variety of security tools built into the Operating System. The main tools were Suricata, Zeek, and a built-in SIEM to collect log traffic. Additionally, it is recommended use PFsense to direct traffic and Squid proxy for web traffic logging and SSL/TLS inspection. Due to the technical limitations of the virtual infrastructure this architecture had to change.

From there Pfsense was implemented with Suricata and Squid proxy as part of that implementation with Splunk as the SIEM.

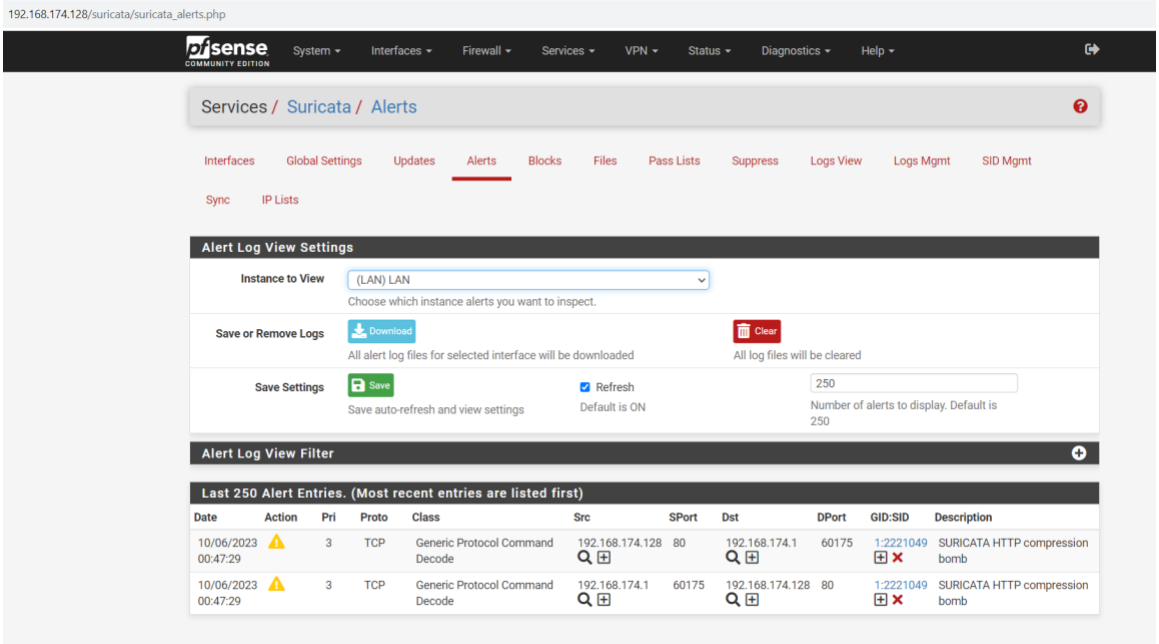


Figure 5. PFsesnse with Suricata

Squid was configured with SSL bump to allow for SSL/TLS inspection. Initial rules were created within Suricata to detect Social Security Numbers being inputted into chat.openai.com. Even though it appeared to be decrypting the packets based on the Squid logs, the cleartext traffic that was input into ChatGPT could not be found. Further research indicated that the headers may have been decrypted but the body of the packets was not, which is where the input text is located.

Next, traffic was captured within Wireshark and decrypted using the SSLKeyLog file. The plan was to export the relevant packets from Wireshark and then run them through Suricata to check if the alert rules were firing as expected.

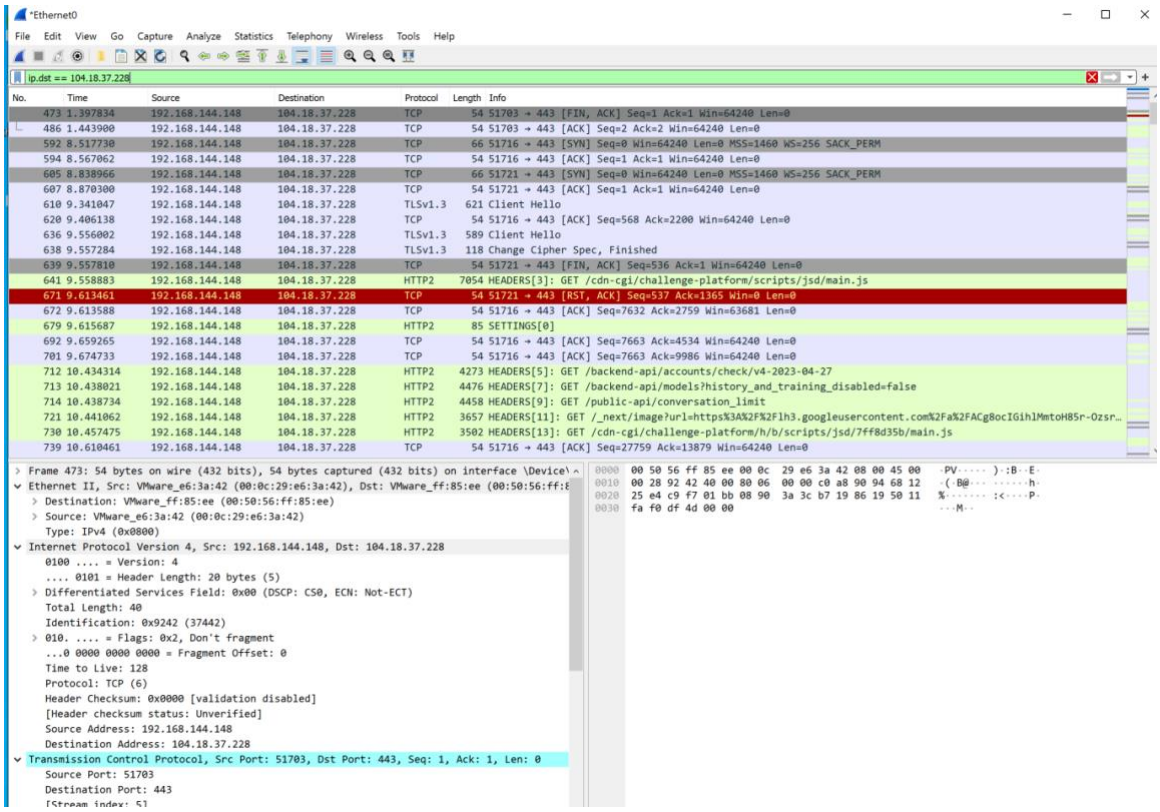


Figure 6. Wireshark with Decrypted Traffic

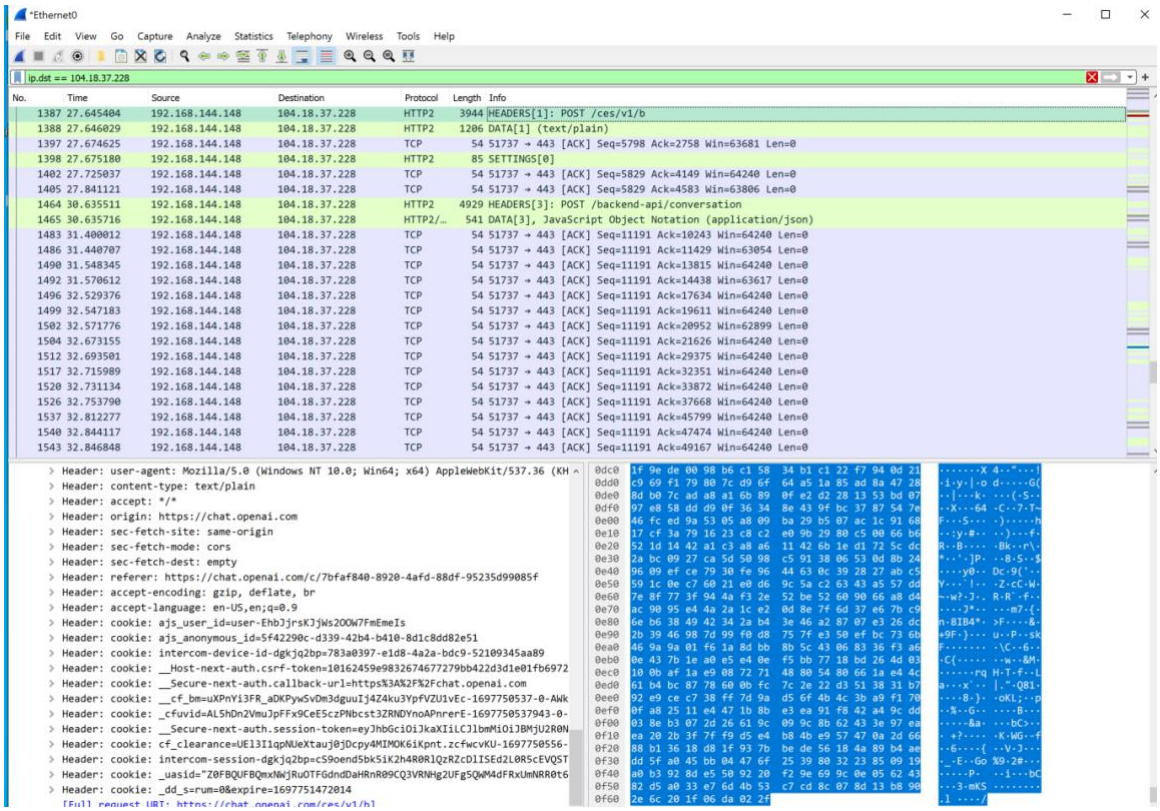


Figure 7. Wireshark – Only Headers Decrypted

Unfortunately, this solution had similar issues as the previous environment. The GET and POST requests were visible indicating the packets were decrypted, but the input text could not be found. To ensure that the QUIC protocol was not causing the issue, it was disabled within the Chrome browser.

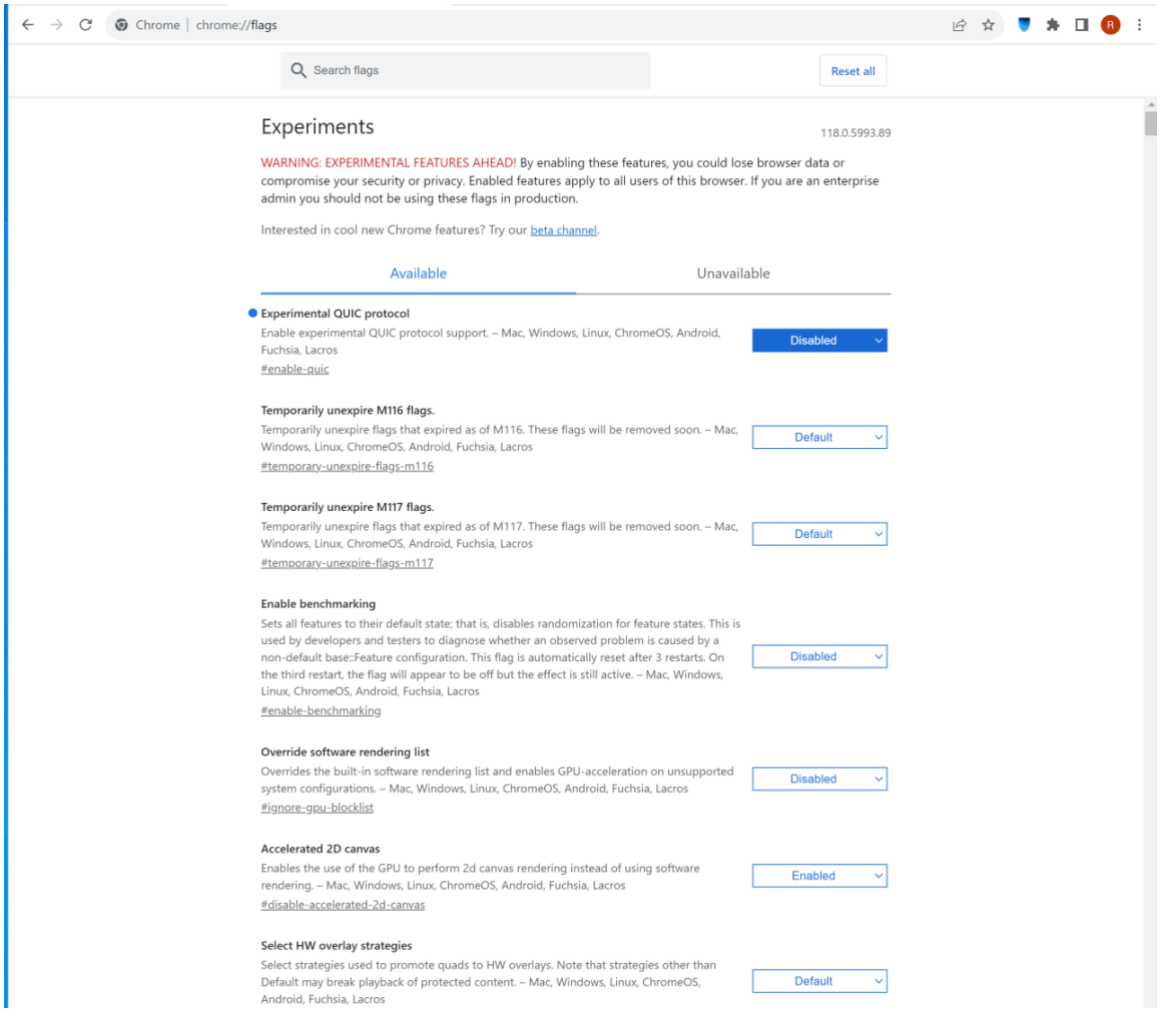


Figure 8. QUIC Disabled

The next attempt was to use Fiddler Proxy or Burpsuite Proxy to record the traffic, forward the logs to Splunk, and create an alert in Splunk to at least detect suspicious text being input into ChatGPT. This, however, failed as well. 403 HTTP status codes were received when SSL/TLS inspection was turned on for these browsers.

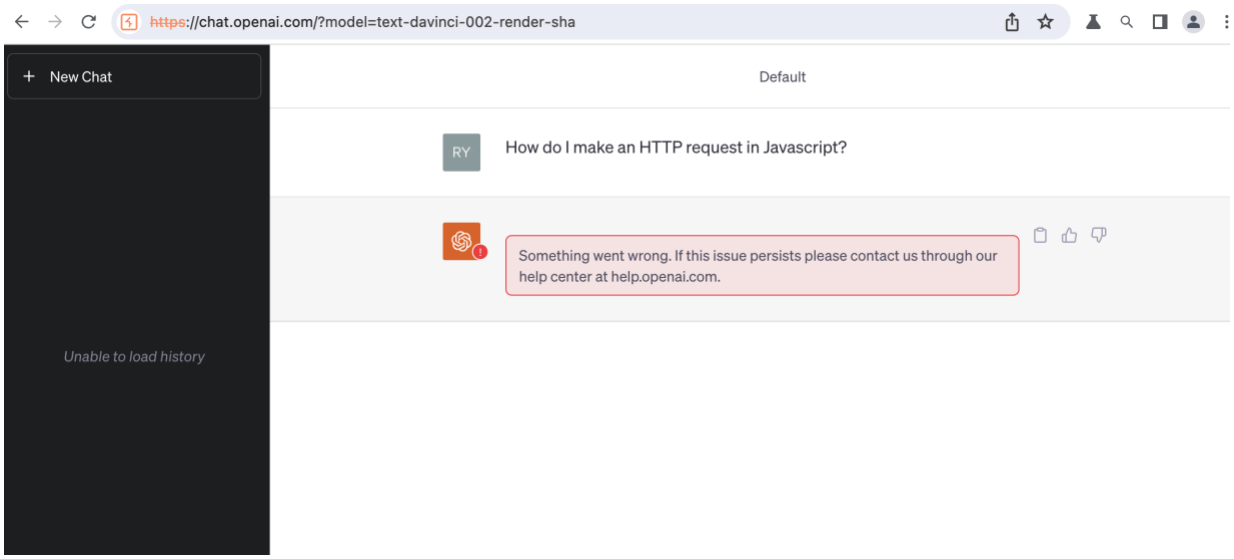


Figure 9. Burpsuite Browser Error

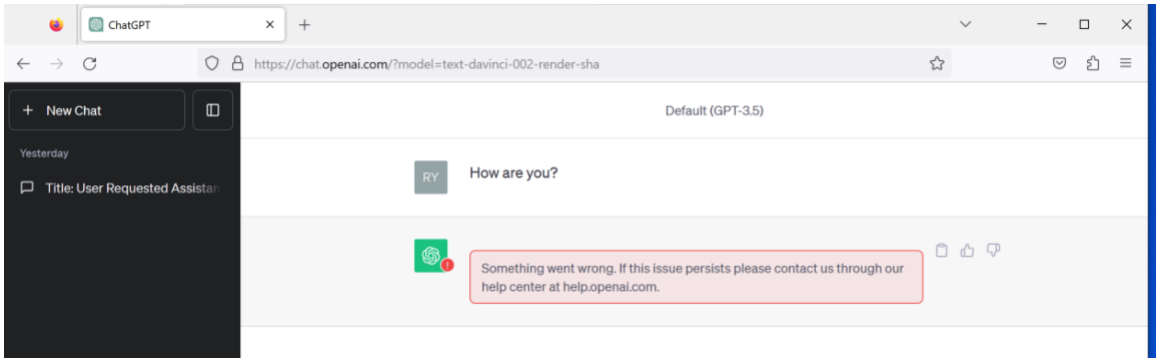


Figure 10. FireFox Browser Error

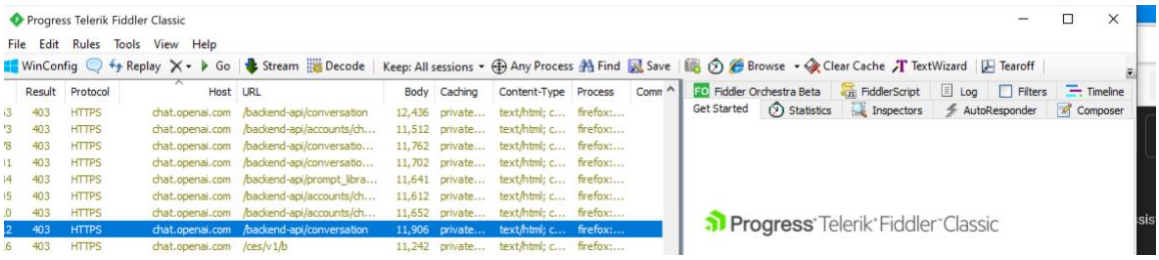


Figure 11. Fiddler HTTP Status 403

Other researchers had also seen this issue, but only in Chromium browsers. The Firefox web browser was downloaded, and it was confirmed there was still an issue.

Ryan McDonald, j.ryan.mcdonald@gmail.com

Further research indicated it is related to TLS 1.3, so the browser was forced to use TLS 1.2. That did not resolve the issue and errors continued from the ChatGPT application.

After all other choices had failed, PolarProxy was installed and configured. This is a useful tool in that it has great support and documentation to help users get set up quickly. Some of the best features allow it to decrypt traffic and output to a packet capture file easily. This makes it easy to load into Wireshark and determine if it is performing as expected. PolarProxy was able to decrypt the ChatGPT traffic to allow Suricata to detect suspicious usage. Figure 12 below shows text entered into ChatGPT and then identified in Wireshark.

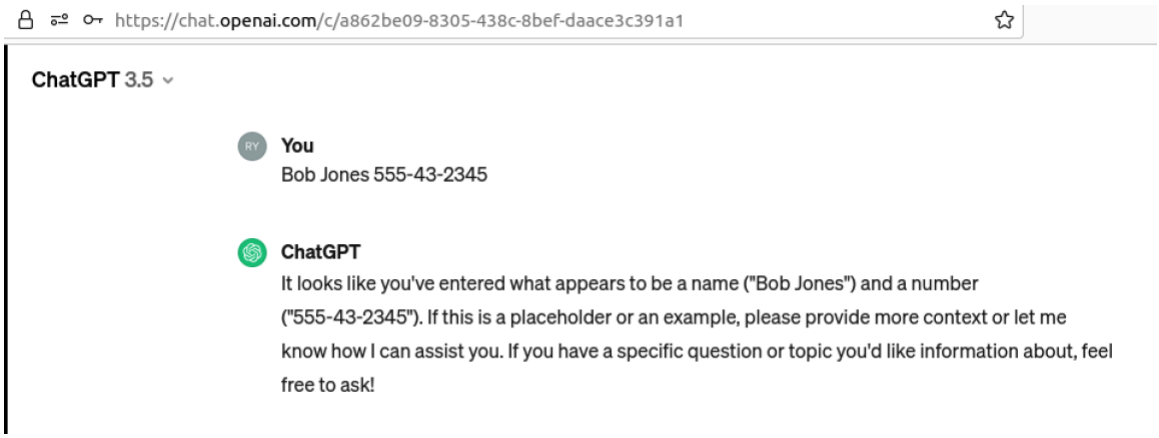


Figure 12. ChatGPT Text

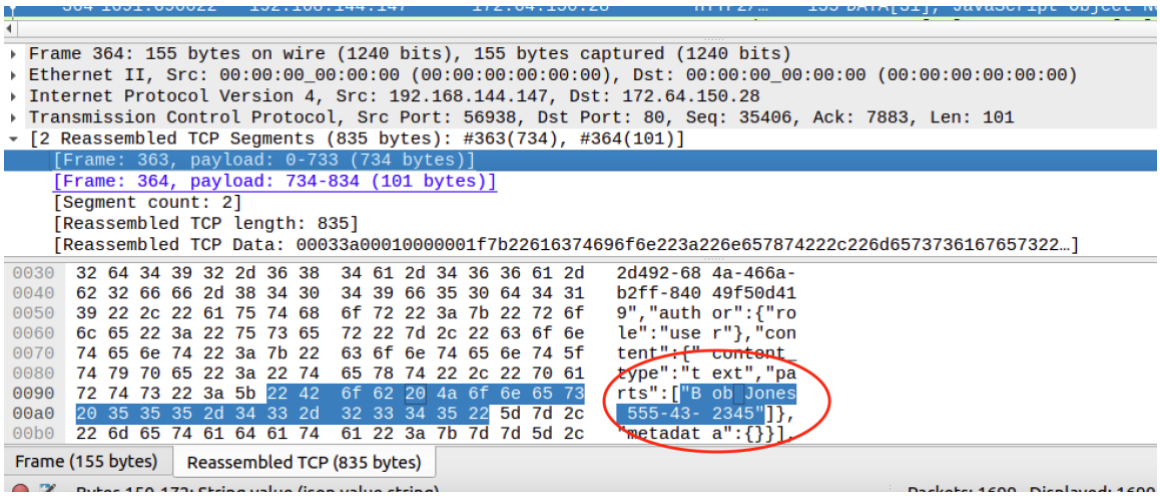


Figure 13. Cleartext in Wireshark

Once that SSL/TLS inspection was successful, the next step was to test Suricata rules against the traffic to see if an alert could be created. Suricata was installed on Ubuntu and alert rules were created. Custom alert rules were placed in the /etc/suricata/rules/local.rules file for testing. Next, the suricata.yaml was configured to set the $HOME_NET and $EXTERNAL_NET variables. A simple rule to alert on any traffic to the ChatGPT destination IP (*alert http2 any any -> 172.64.150.28 any (msg:"Detect HTTP/2 traffic to 172.64.150.28"; sid:1000003;))* was created to test if everything was working and was successful.

Next, a rule was created to look for Social Security Numbers within any traffic destined for chat.openai.com.

alert http2 any any -> 172.64.150.28 any (msg:"Detect SSN in decrypted HTTP/2 traffic"; content:"\\d{3}-\\d{2}-\\d{4}"; sid:1000005;)

The rule was able to find the packet that contain the Social Security Number as seen in Figure 14:

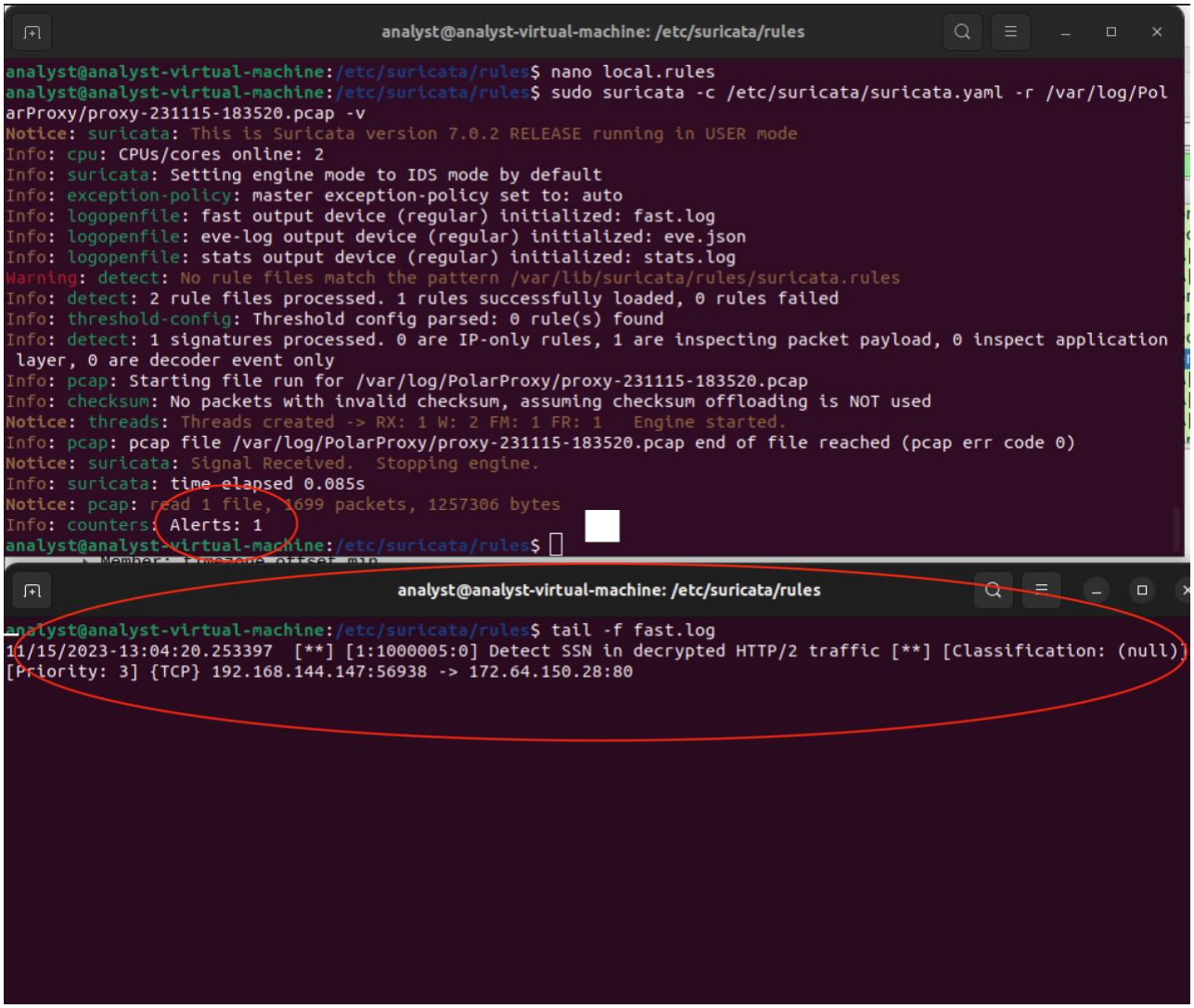


Figure 14. Suricata Alert

Once visibility into the encrypted web traffic was established, dashboards and alerts could be created in the SIEM for further monitoring. The unencrypted web traffic was imported into Splunk and dashboards were created to demonstrate which users are using the service more than others, how often they visit, and even their queries. From this data monitoring, rules and alerts can be created to ensure visibility into the organizations GenAI usage.

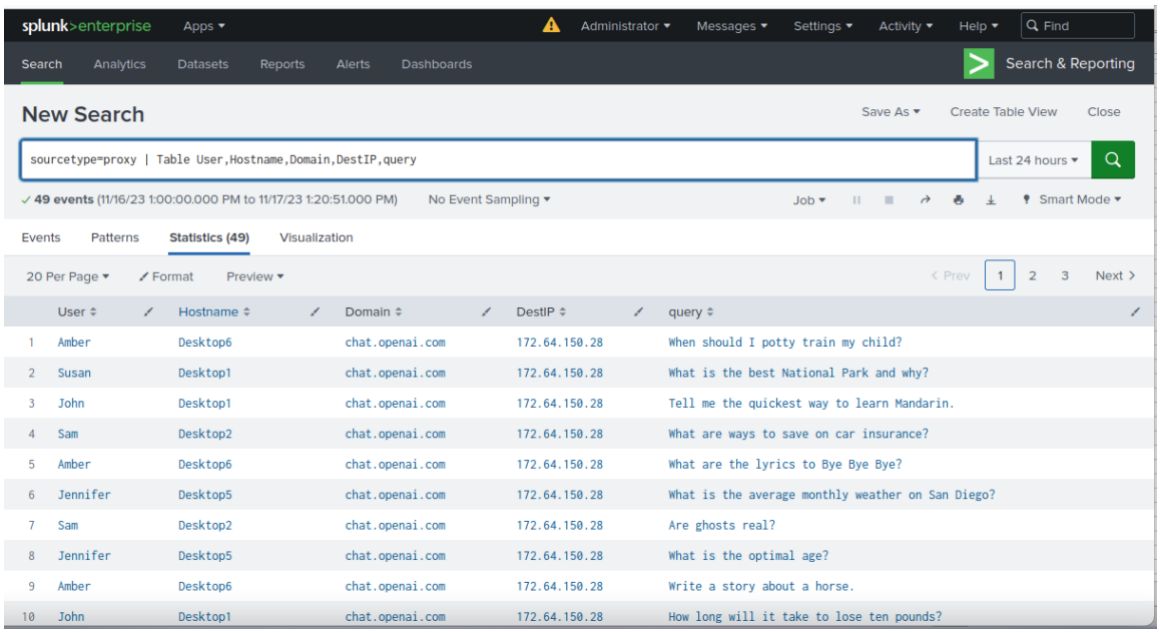


Figure 15. Splunk Dashboard

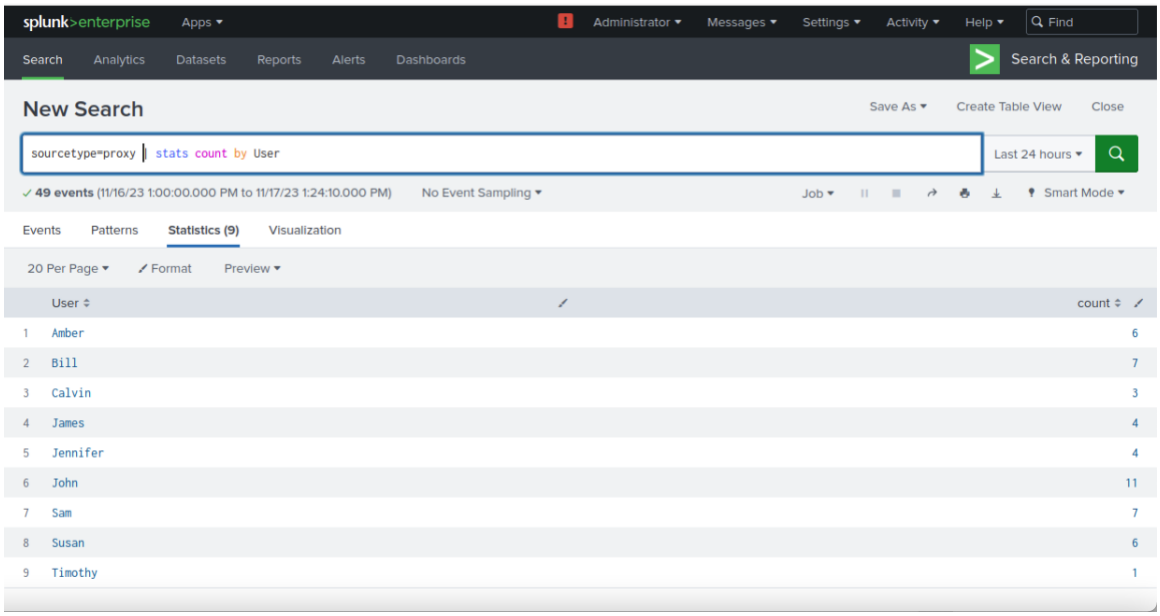


Figure 16. Splunk User Count

The figures above are examples for ChatGPT, but with full web traffic visibility organizations can add other domains as well, such as Google's Bard. However, as more and more GenAI websites come online, this will become more and more difficult to manage with some type of automation built in.

Ryan McDonald, j.ryan.mcdonald@gmail.com

3.4.2. Host-Based Testing

A host-based solution offers many advantages when compared to a pure network solution. Unfortunately, an open-source DLP agent or EDR solution could not be found that would meet our use case. However, a browser extension was discovered that offered potential. LakeraAI seemed to meet the criteria needed and had some functionality already built in. This would save significant time versus trying to code a browser extension from scratch.

The browser extension was written to be used in the Chrome browser. This is a common browser, so that would not be a limitation for many organizations. The installation was easy and can be automated in all company web browsers to simplify the process. As mentioned above, the native capabilities included are being able to block or obfuscate the following data types from being input into ChatGPT:

- Credit card numbers
- Anglophone names
- Email addresses
- Phone numbers
- US street addresses
- US Social Security Numbers
- Secret keys

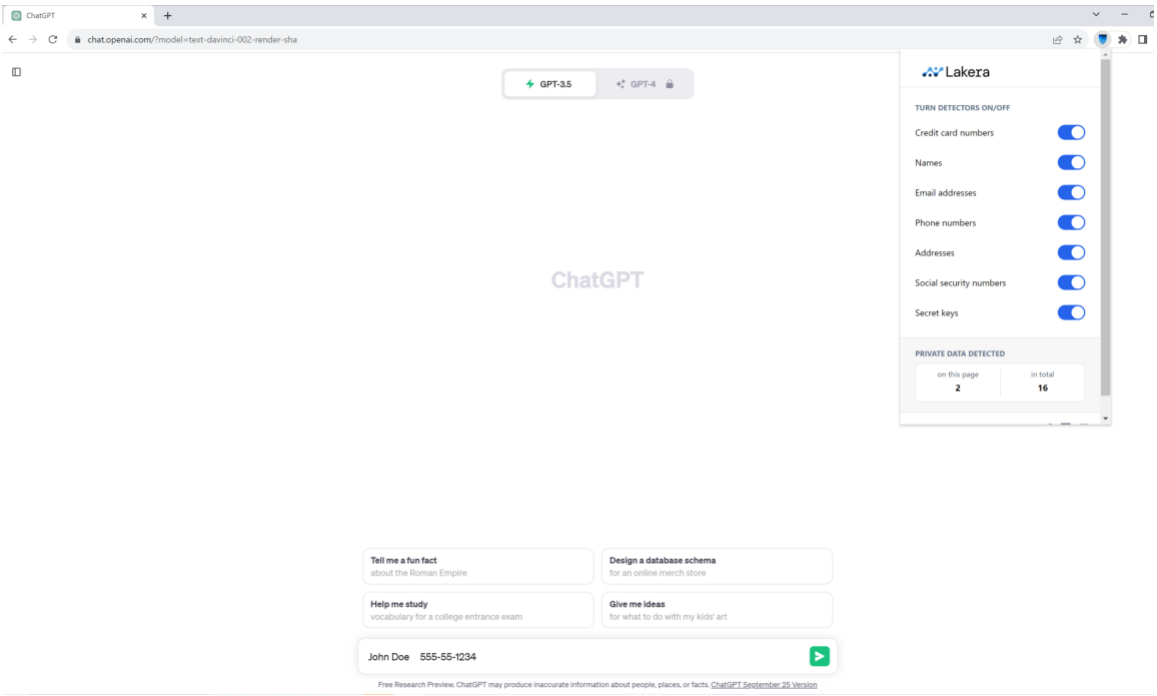


Figure 17. LakeraAI Detectors

After installation, testing the various data types began to see what would happen. Random social security number were inputted, and a pop-up alert was received. More testing was completed with names, addresses, and phone numbers and all alerted as expected.

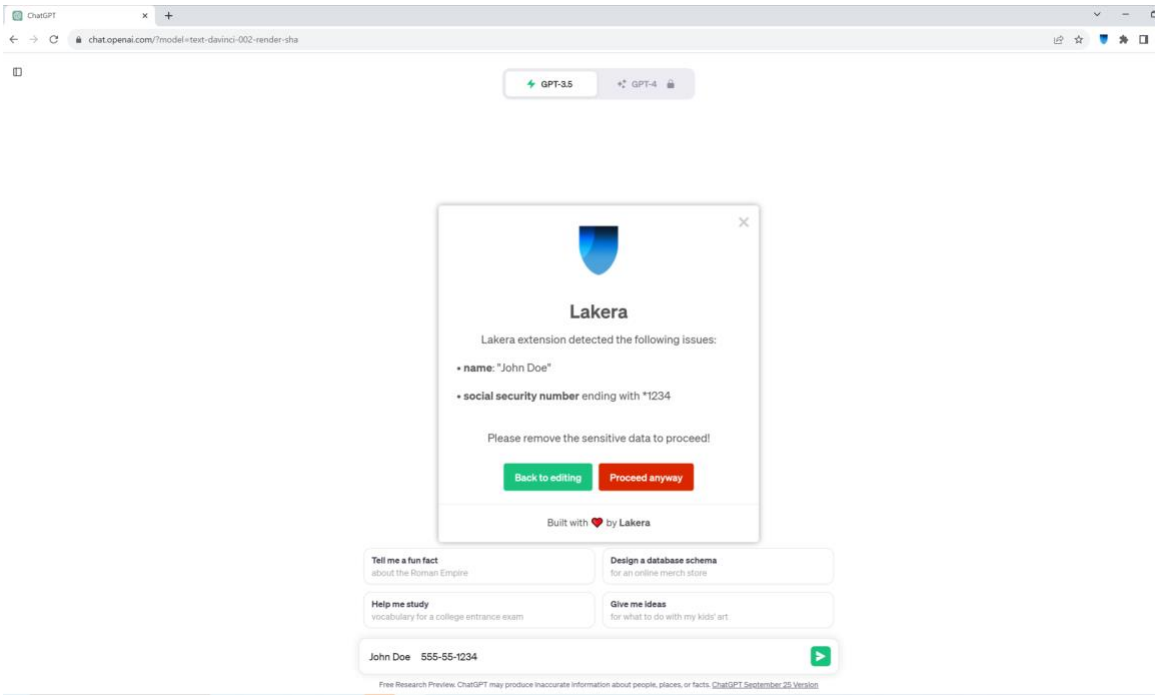
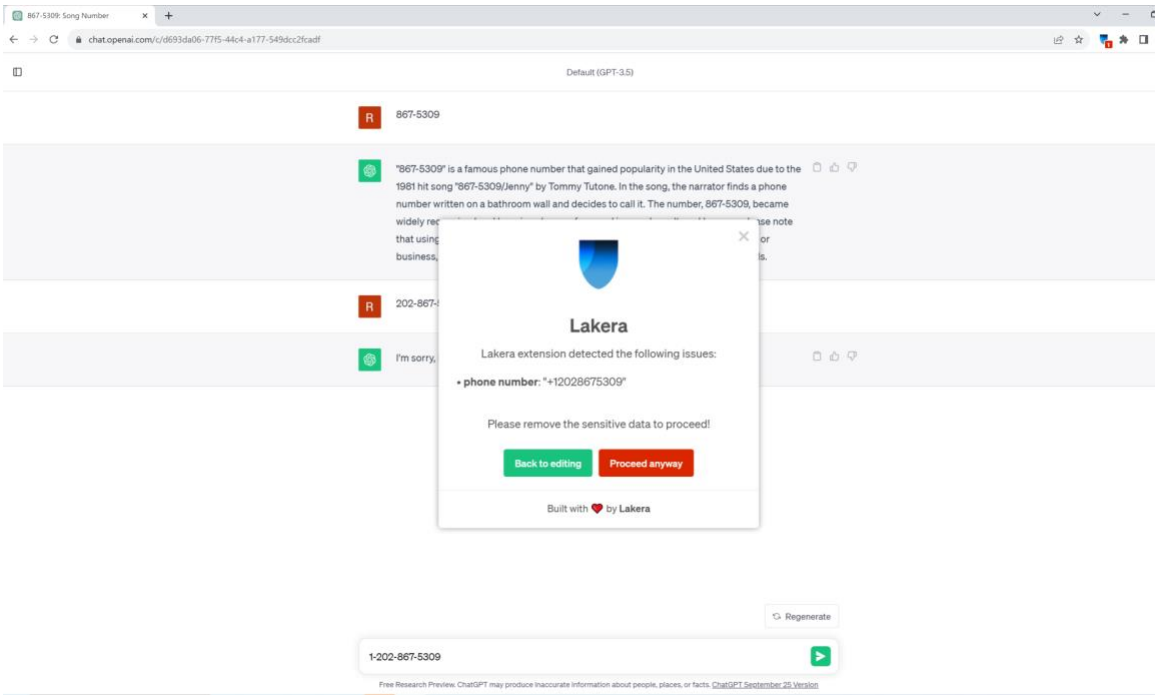


Figure 18. LakeraAI Pop Up Alert

The only issue discovered was with phone numbers. The regular expression needs to be modified because it only alerts on phone numbers when 1 is added in front of the area code.



Ryan McDonald, j.ryan.mcdonald@gmail.com

Figure 19. LakeraAI Phone Number Pop Up Alert

After testing multiple times, the browser extension's code located in LakeraAI's GitHub repository was reviewed to see how it worked behind the scenes and see how it could be customized. Some easy changes would be to change the native icons with a company logo or brand. This would make it look very legitimate and credible with end users. Another thought was to change the popup.html file with one that explained why it was blocked and had a link to the company GenAI usage policy.

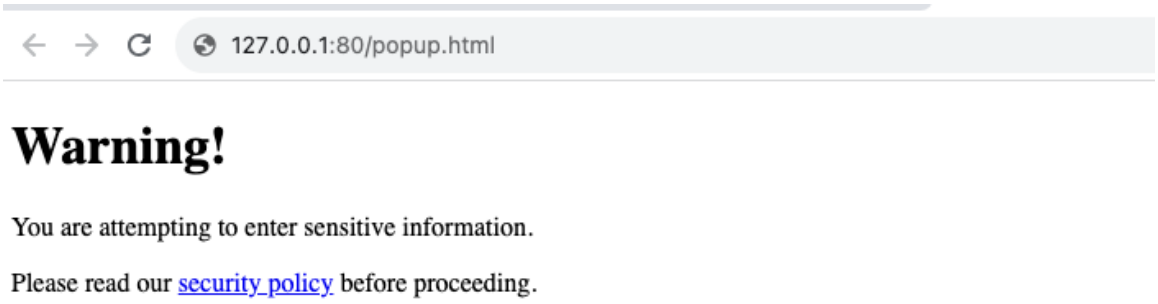


Figure 20. ChatGPT Prompt Text

Overall, this was a great success. There are many customizations that can be made to improve the product. For example, a regular expression could be added that would match specific code, such as Python or JavaScript. Another customization would be to block users who were trying to input specific keywords that may be related to special programs within our organization. These use cases would apply to many organizations and would provide an effective GenAI DLP capability.

While there are many benefits to the host-based solutions there are also some limitations. First, these are only specific to ChatGPT. Many GenAI websites exist where these protections would not help, such as Google's Bard. Also, this requires the use of Chrome browsers, which, while it is a very popular browser, many organizations are

hesitant to force users only to use one specific browser. Sometimes applications only work with specific browsers, which could complicate the issue.

4. Recommendations and Implications

While organizations can allow the usage of ChatGPT and other GenAI tools without introducing undue risk to their organization, some considerations will need to be made before introducing them into a production environment. These considerations and how this research can be improved upon, will be discussed in the following sections.

4.1. Recommendations for Practice

It is important to understand that implementing any inline IPS solution will have many challenges. Adding SSL/TLS inspection adds to these challenges. Some challenges include blocking legitimate business traffic, network delays, and privacy risks associated with decryption. Using network or host-based protections will require continued maintenance to keep up with new GenAI web applications and websites.

4.2. Implications for Future Research

With the change to a more remote and distributed working environment, having a pure network solution will not be a comprehensive solution. For these use cases, it is important to push the protections to the endpoint as much as possible. Any research that will add to this capability, especially from an open-source perspective, would be extremely helpful to the security community going forward.

5. Conclusion

GenAI technologies are completely changing how companies do business. It is equipping employees with tools to do their job better and faster. As more users interact and use these technologies, they will get better and more accurate. Because of this growing prevalence, companies must be aware of the risks these tools introduce as well as the benefits (Addington, 2023). The risks are small compared to benefits but still need to be reduced and mitigated where possible.

There are commercial solutions available to help with these risks (Sridhar, 2023), but for many organizations the ability to purchase these tools is too high. Thanks to the open-source community there are other cost-effective, viable solutions that organizations can consider. Along with these technical solutions, companies must also make sure they have policies and training in place for users to know what is acceptable when interacting with these technologies and what can add additional risk. Having wholistic administrative and technical controls around GenAI can go a long way to reducing risk while taking advantage of the benefits of this transforming technology.

References

Addington, S. (2023, May 9). *CHATGPT: Cyber security threats and countermeasures*. SSRN. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4425678

Derner, E., & Batistič, K. (2023, May 13). *Beyond the safeguards: Exploring the security risks of chatgpt*. arXiv.org. https://arxiv.org/abs/2305.08005

DeRose, A. (2023, May 19). *These companies have banned or limited CHATGPT at work*. HR Brew. https://www.hr-brew.com/stories/2023/05/11/these-companies-have-banned-chatgpt-in-the-office

Enterprise DLP administrator's guide. How Enterprise DLP Safeguards Against ChatGPT Data Leakage. (2023, June 23). https://docs.paloaltonetworks.com/enterprise-dlp/enterprise-dlp-admin/configure-enterprise-dlp/enterprise-dlp-and-ai-apps/how-enterprise-dlp-safeguards-against-chatgpt-data-leakage

Fearn, N. (2023, June 20). *CHATGPT is creating a legal and compliance headache for Business: Computer Weekly*. ComputerWeekly.com. https://www.computerweekly.com/feature/ChatGPT-is-creating-a-legal-and-compliance-headache-for-business

Sridhar, M. (2023, April 13). Netskope DLP use cases for ChatGPT [web log]. Retrieved June 28, 2023, from https://community.netskope.com/t5/Inside-Netskope-Security/Netskope-DLP-use-cases-for-ChatGPT/td-p/3992.

Sahu, A. (2023, February 22). *A quick look into CHATGPT's network traffic*. Keysight. https://www.keysight.com/blogs/tech/nwvs/2023/02/21/a-quick-look-into-chatgpts-network-traffic

Tilley, A. (2023, May 19). *WSJ News Exclusive | Apple restricts employee use of CHATGPT, joining other companies wary of leaks*. The Wall Street Journal. https://www.wsj.com/articles/apple-restricts-use-of-chatgpt-joining-other-companies-wary-of-leaks-d44d7d34

Appendix
List of Figures