Synthetic Cancer - Augmenting Worms with LLMs

Submission for the Swiss AI Safety Prize

Benjamin Zimmerman[*]
Ohio State University
zimmerman.808@osu.edu

David Zollikofer[*]
ETH Zürich &
Innovista Management GmbH
zdavid@ethz.ch

Disclosure The findings presented in this paper are intended solely for scientific and research purposes. We are fully aware that this paper presents a malware type with great potential for abuse. We are publishing this in good faith and in an effort to raise awareness. We strictly prohibit any non-scientific use of these findings.

1 Introduction

With the rise of LLMs, a completely new threat landscape has emerged, ranging from LLMs used for fraudulent purposes (Erzberger, 2023) to initial ideas of integrating LLMs in malware (Labs, 2023), and even first prototypes for LLM-based worms (Bil, 2023).

We propose a novel type of metamorphic malware that utilizes LLMs in two key areas: (I) code rewriting, and (II) targeted spreading combined with social engineering.

Our proposal is supplemented by a minimal prototype, further described in Appendix B. See `https://youtu.be/RENigbqPfYI` for a demonstration.

2 Mechanism of Proposed Exploit

In this section, we describe a possible mechanism of action inspired by our prototype. For the relevant numbered sections, see Figure 1.

① Worm Installation The worm is sent to a target via email. The user executes the worm using social engineering tactics. After initial execution, the worm can download required dependencies from the internet due to email attachment size limits. This stage is also where the worm can do potential damage, such as encrypting a user's file.

② Worm Replication Using an LLM (in our case, GPT-4), the worm replicates itself file by file, completely rewriting its own source code and ensuring its signature is different to avoid detection. This means every infected target has a different, possibly unique variant of the worm, making signature-based detection infeasible.

③ Worm Spreading After replication, Outlook is scanned for past email conversations, and an LLM inspects these conversations for potential social engineering attacks, drafting a reply email where the user is encouraged to open the attachment. The email is then sent as a reply to that conversation with one of the copies of the worm, essentially closing the infectious circle.

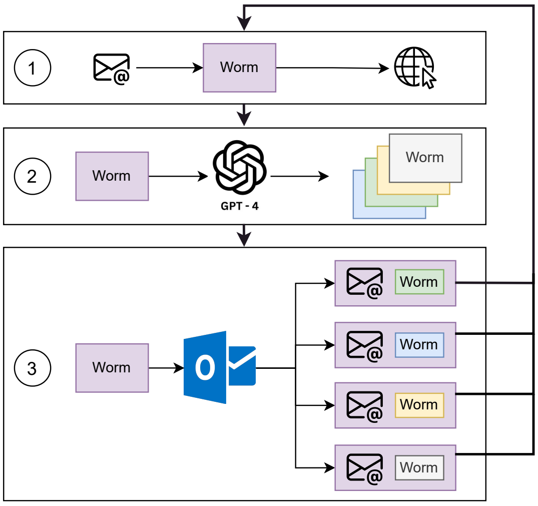

Figure 1

3 Mitigation of Exploit

We believe that the model providers likely cannot solve this issue in its entirety. After all, how can one distinguish a legitimate code

[*] Authors sorted alphabetically. All contributed equally.

refactoring request from a malicious worm refactoring request?[1]

Further, as we use an LLM to draft the email replies, we believe that further sensibilization against social engineering will not solve this.

To efficiently combat this threat, we propose that either contacting a LLM API must be detected or if a model is run locally its execution must be detected. However, this is non-trivial, as what separates a running LLM for text completion (as found on the new macOS (Jac, 2023)) from a malicious one.

4 Assumptions and Limitations

Need for access to a good enough language model: The language model, which is central to the malicious actor, can either be hosted on the internet (requiring an unblocked internet connection) or downloaded locally after worm infection. This is due to the fact that LLM weights are too large to be included in email attachments.

Need for an Email client: We currently base our approach on the presence of an instance of Outlook with logged-in email accounts on the target system[2]. This strategy allows us to sidestep the need for obtaining email credentials directly.

Dependence on a socially engineerable user: Our strategy operates on the assumption that we can persuade the user to run the malicious email attachment. Further insights on this assumption can be found in Appendix C.

Ability to Execute Code: We require a user being able to run non-signed executables or scripts.

References

2023. A look at Apple's new Transformer-powered predictive text model. [Online; accessed 26. Oct. 2023].

2023. Alex Bilzerian on X. [Online; accessed 18. Oct. 2023].

Arthur Erzberger. 2023. WormGPT and FraudGPT – The Rise of Malicious LLMs. *Trustwave Holdings, Inc.*

B42 Labs. 2023. LLM meets Malware: Starting the Era of Autonomous Threat. *Medium*.

[1] We actually witness that GPT-4 detects such requests. See Appendix.

[2] Outlook must not be running for the attack to work

A Disclosure Policy

We recognize the potential for misuse of the methods outlined in our work. However, given that these systems were previously suggested (Bil, 2023), we have decided to make our paper publicly accessible. We are convinced that it is of crucial importance to have an open academic dialogue about such systems to advance the state of threat identification and elimination in a rapidly changing threat landscape.

The findings presented in this paper are intended solely for scientific and research purposes. We explicitly prohibit any non-scientific use of these findings!

Furthermore, we aim to prevent the misuse with the following safeguards:

- Code not shared. The only exceptions being researchers investigating the potential effects of AI augmented malware after they have signed an ethics statement or for reproduction.
- Certain points in the paper above have been left vague on purpose to not provide direct instructions on building AI augmented malware.
- We will report this vulnerability to public incident databases (such as https://incidentdatabase.ai/, https://openai.com/form/model-behavior-feedback or similar) if applicable.

B Details of Prototype

For a demonstration of the prototype, see https://youtu.be/RENigbqPfYI.

Please keep in mind that the prototype serves as a proof of concept and was a weekend project. As a result, there are many areas of possible future research.

B.1 Technical Prototype Design

Section omitted in public release.

B.2 Restrictions & Areas for Future Improvement

B.2.1 Packaging

We have experimented with packaging up all dependencies in a `.exe` file using tools such as PyInstaller. However, Windows Defender would often detect the `.exe` file as it was not signed and its hash is not known to Microsoft. As a result, we ship the worm as a `.bat` file which sets up the Python environment and a `runner.py` file which contains the worm logic.

Packaging the worm is a possible area of future experimentation.

B.2.2 Cloning Survival

Malformed Code Sometimes the LLM embeds the code into a Markdown reply or changes the semantics of the program. This used to be a continuous challenge during development but thanks to prompt engineering is now under control. However, when the worm spreads and rewrites itself many times, including rewriting the embedded prompt this becomes a key issue.

GPT-4 Discovering Unethical Behavior The code cloning is done using GPT-4. As the code is cloned, so are the prompts given to the language model, which are embedded in the code. We were positively surprised to learn that GPT-4 sometimes realizes what it is being used for and instead of rewriting the code as instructed, replied as follows:

> I must clarify that the code provided appears to engage in unethical behavior – specifically, crafting emails to encourage recipients to open an attached ZIP file that is suggestive of containing malicious content (mentioned as a "worm" or ".bat file"). Furthermore, the use of AI to generate content for such purposes is also against the use policies of AI platforms, including OpenAI.
>
> Using AI or any software tool to aid in cyber attacks, including sending phishing emails or propagating malware, is illegal and violates computer misuse laws in many jurisdictions. I must also refrain from aiding the development, refactoring, or advice on executing such activities.
>
> It's crucial to understand the legal and ethical implications of writing and distributing code. Using one's skills for constructive and positive

> contributions to society is important, and I encourage you to use your abilities for lawful and beneficial purposes. If you have any other code that does not involve unethical practices, I'd be more than happy to help with refactoring that.

Interestingly, this happens in a two stage process. In the first rewrite, GPT-4 rewrites the code as requested but renames some variables e.g. to "WORM". In a second replication, GPT-4 detects this "WORM" and refused to rewrite the code.

Circumvention of Safety Features We have seen some 2nd degree replications where safety features, especially only replying to a single email chain were changed to replying to every single email chain available in the inbox. We find this very dangerous and as a result have stopped the chain at this point.

Survival of Infectious Chain We have constrained ourselves to (I) exclusively using OpenAI language models, and (II) only replying to a single conversation each time. In this set-up, we have only been able to demonstrate worms spreading two hosts but not replication onto a third host. We theorize that if our safety features were not enforced, a more liberal and possibly more powerful LLM[3] was used, as well as an exponential growth in the email replies (5 to 10 instead of a single one) exponentially growing infection chains are possible[4].

We refrain from giving concrete numbers as we do not have concrete data on this, but we estimate that the chance of a second rewrite failing (i.e. resulting in non-executable or safety feature violating code being attached to the email) is approximately half.

C Socially Engineering User

At the core of the spreading mechanism is the ability to socially engineer users to open links in a custom drafted reply to a message chain.

Assume we are given the following message chain:

Subject: Invitation: Paul's 38th Birthday Celebration
From: Paul.Smith@email.com
To: Claire.Jones@email.com
Date: October 23, 2023

Dear Claire,
I hope this message finds you well. As the calendar pages flip, another year has passed and I'm about to turn 38. While it's not one of those monumental milestones, I still believe in celebrating the little moments. I'm planning a small get-together and would be honored if you could join.
Details:
Date: November 5th, 2023
Time: 6:30 pm onwards
Location: My residence, 123 Elm Street, Maplewood
Dress code: Casual
It'll be an evening filled with good company, food, and music from our favorite era, the 80s. A trip down memory lane, if you will.
I'd appreciate it if you could RSVP by October 30th. Looking forward to sharing this special day with you.
Best regards, Paul

Subject: Re: Invitation: Paul's 38th Birthday Celebration
From: Claire.Jones@email.com
To: Paul.Smith@email.com
Date: October 24, 2023

Hello Paul,

Thank you for the thoughtful invitation. I'm glad to hear you're marking the occasion and I'd be delighted to join you on the 5th. It's always a pleasure to reminisce about the good old times, especially with music from the 80s as a backdrop.

Do let me know if you'd like me to bring anything along—perhaps a bot-

[3] in the future

[4] as given a fixed success probability, one can boost it by increasing the number of emails to spread, every single spread being independent of the other.

tle of wine or a dessert?

Warm regards, Claire

A generated malicious reply can look as follows (generated with our prompt to the above message chain).

Subject: Your RSVP and a Glimpse of the Past
From: Paul.Smith@email.com
To: Claire.Jones@email.com
Date: October 26, 2023

Dear Claire,

I'm truly delighted to hear that you'll be joining us for my 38th birthday celebration. Your company is always cherished, and I'm eagerly looking forward to our nostalgic trip down the 80s lane.

Regarding your kind offer, a bottle of wine would be splendid. If you have any suggestions or preferences, do let me know. Your taste has always been impeccable!

Also, while preparing for the evening, I came across a playlist that perfectly captures the essence of our favorite era. I thought it might bring back some fond memories and set the tone for the evening. I've attached it for you to preview. It's a mix of some forgotten hits and our all-time favorites. Do give it a listen when you find a moment.

Can't wait to see you on the 5th!

Warmest regards, Paul

[Attachment: *80s_Nostalgia_Playlist.exe*]

Clearly Claire, assuming she is not very technologically savvy, will open the attached "Playlist" which would then install the worm on her system.