New Technique Detected in an Open-Source Supply Chain Attack

Archived: 2026-04-05 22:38:19 UTC

In a recent attack campaign, cybercriminals were discovered cleverly manipulating GitHub's search functionality, and using meticulously crafted repositories to distribute malware.

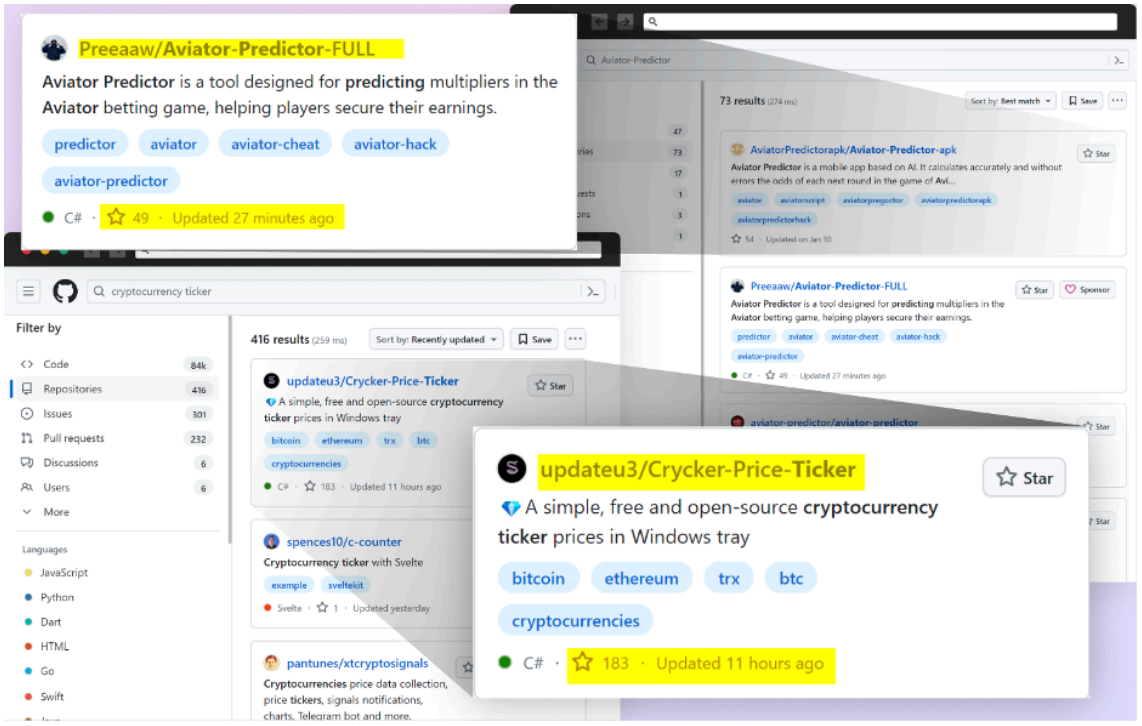
Key Points

- GitHub search manipulation: Attackers create malicious repositories with popular names and topics, using techniques like automated updates and fake stars to boost search rankings and deceive users.
- Malicious code is often hidden within Visual Studio project files (.csproj or .vcxproj) to evade detection, automatically executing when the project is built.
- The attacker had set up the stage to modify the payload based on the victim's origin, checking specifically if the victim is based in Russia. At this point, we don't see this ability activated.
- The recent malware campaign involves a large, padded executable file that shares similarities with the "Keyzetsu clipper" malware, targeting cryptocurrency wallets.
- The malware establishes persistence on infected Windows machines by creating a scheduled task that runs the malicious executable daily at 4AM without user confirmation.
- Developers should be cautious when using code from public repositories and watch for suspicious repository properties, such as high commit frequencies and stargazers with recently created accounts.

Exploiting GitHub's Search Functionality

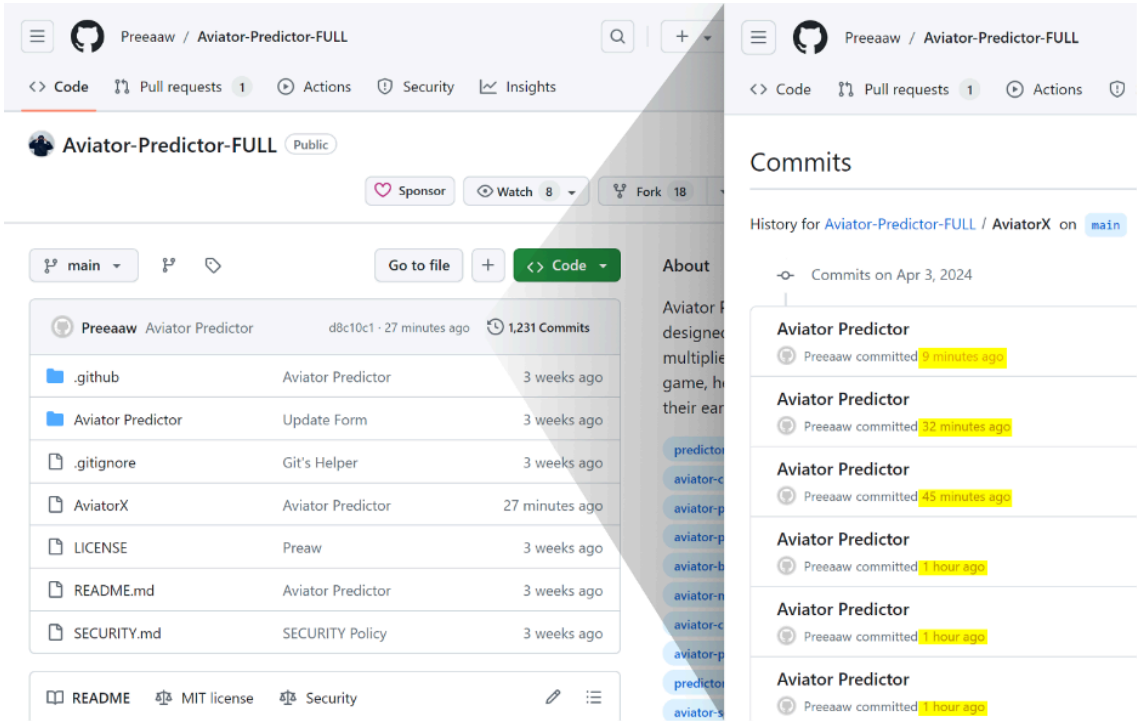
Our recent findings reveal a threat actor creating GitHub repositories with names and topics that are likely to be searched by unsuspecting users. These repositories are cleverly disguised as legitimate projects, often related to popular games, cheats, or tools, making it difficult for users to distinguish them from benign code.

To ensure maximum visibility, the attackers employ a couple of clever techniques that consistently place their malicious repositories at the top of GitHub search results.



Automatic Updates

By leveraging GitHub Actions, the attackers automatically update the repositories at a very high frequency by modifying a file, usually called “log”, with the current date and time or just some random small change. This continuous activity artificially boosts the repositories’ visibility, especially for instances where users filter their results by “most recently updated,” increasing the likelihood of unsuspecting users finding and accessing them.

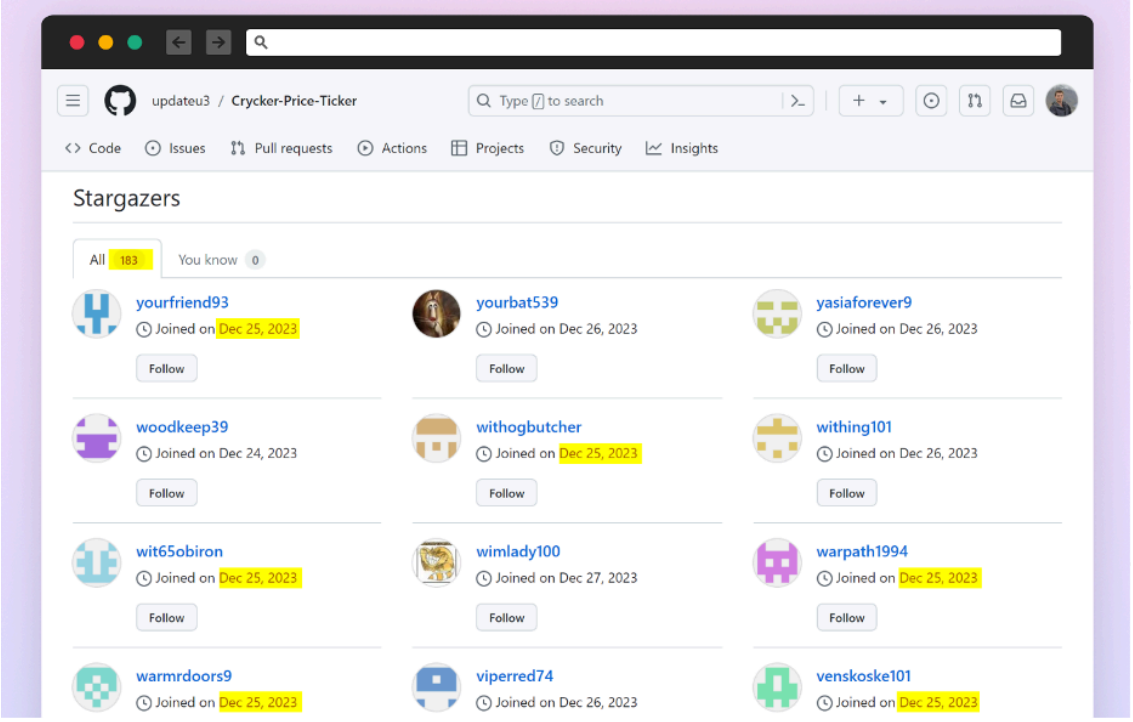


Faking Popularity

While automatic updates help, the attackers combine another technique to amplify the effectiveness of their repo making it to the top results.

The attackers employed multiple fake accounts to add bogus stars, creating an illusion of popularity and trustworthiness. This artificially boosts the repositories' visibility further, especially for instances where users filter their results by "most stars."

In contrast to past incidents where attackers were found to add hundreds or thousands of stars to their repos, it appears that in these cases, the attackers opted for a more modest number of stars, probably to avoid raising suspicion with an exaggerated number.



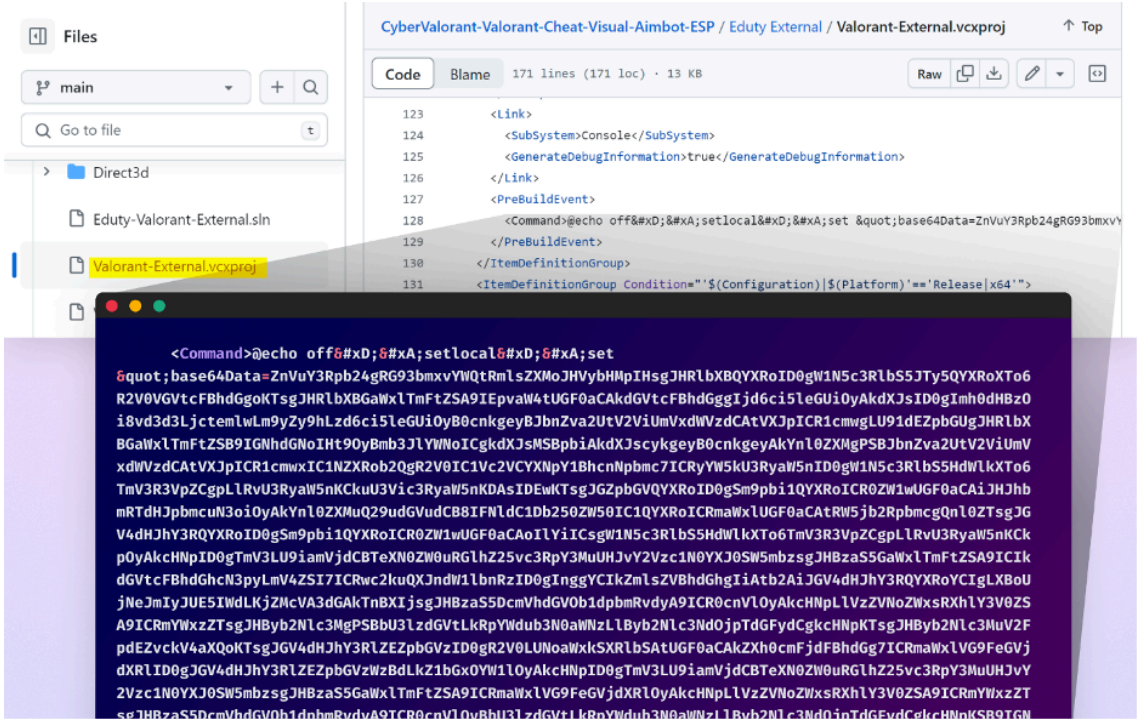
Many of the stargazers are created on the same date. A red flag for fake accounts.

This social engineering technique is designed to manipulate users into believing that the repository is widely used and reliable, preying on the inherent trust users place in highly-starred repositories.

Unsuspecting users, often drawn to the top search results and repositories with seemingly positive engagement, are more likely to click on these malicious repositories and use the code or tools they provide, unaware of the hidden dangers lurking within.

For a deeper dive into the tactic of fake stars, check out our recent blog that explores this manipulation technique in greater detail.

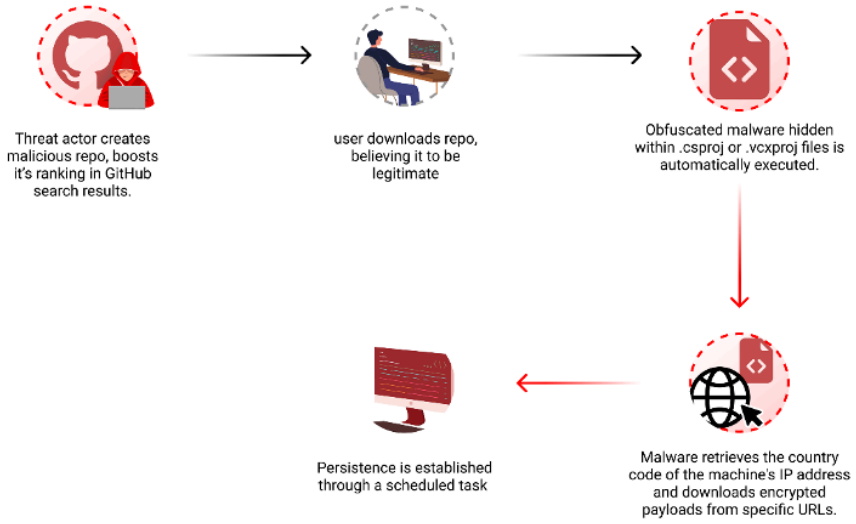
Hidden Malware in Project Files



The attackers conceal their malware primarily as obfuscated code deep within the **.csproj** or **.vcxproj** files of the repository (files commonly used in Visual Studio projects) to decrease the chances of the average user detecting it unless they proactively search for suspicious elements.

However, it's worth noting that there have been a small number of other detected repos that contained different malware within other files.

Technical Analysis of the Common Malicious Payload



The malicious script is embedded within a pre-build event of a Visual Studio project file (.vcxproj) and is designed to be executed automatically during the build process. The script consists of two main parts:

1. A batch script that sets up the environment and executes a VBScript file.

2. A base64-encoded PowerShell script that is decoded and executed by the VBScript file.

The batch script creates a temporary directory, generates a VBScript file, and decodes the base64-encoded PowerShell script. It then executes the decoded PowerShell script and cleans up the temporary files.

The decoded PowerShell script performs the following malicious actions:

1. Retrieves the country code of the machine's IP address, determining whether the machine is based in Russia.
2. Downloads content from specific URLs based on the country code (content is continuously updated by the attacker)
3. Downloads encrypted files from each URL, extracts them with a predefined password, and executes the extracted files.

The script also employs error handling to silently catch exceptions and continue execution.

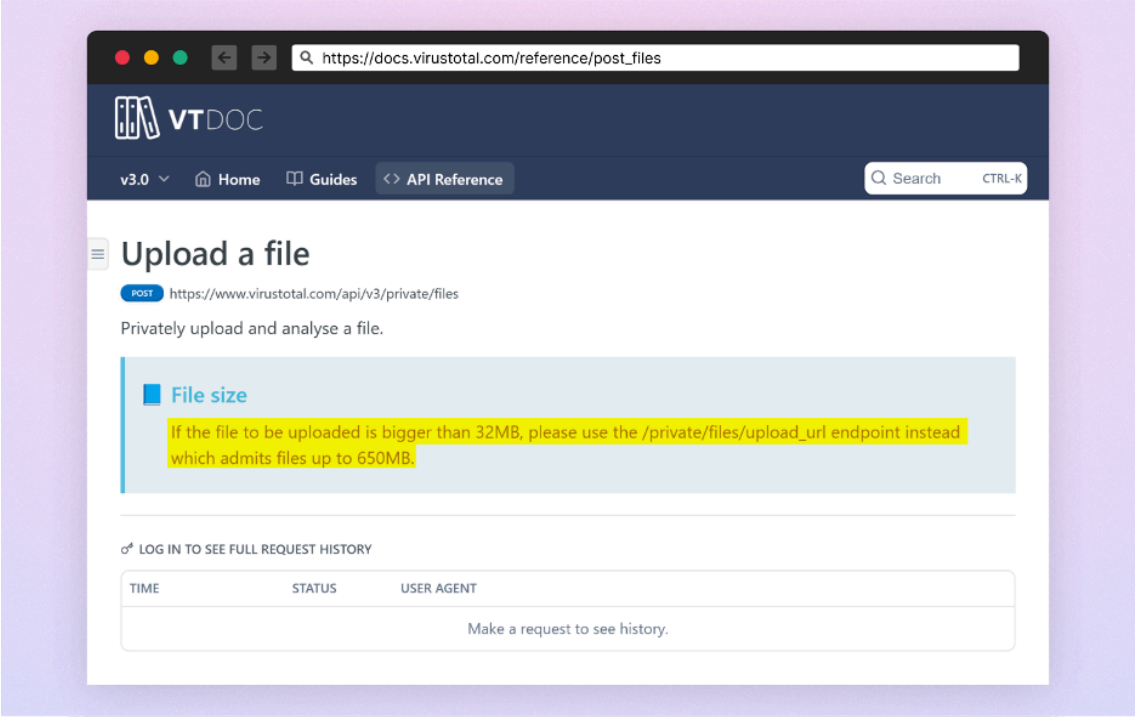
Active Campaign

On April 3rd, the attacker updated the malicious code within one of their repositories, pointing to a new URL that downloads a different encrypted .7z file containing an executable named feedbackAPI.exe.

The attacker had padded the executable with many zeros, a technique used to artificially boost the file size. Due to this padding, the file size exceeded the threshold of many security solutions, VirusTotal being a notable one, preventing the possibility of it from being scanned. According to VirusTotal's documentation,

"*If the file to be uploaded is bigger than 32MB, please use the /private/files/upload_url endpoint instead which admits files up to 650MB.*"

The padded feedbackAPI.exe file was 750MB in size, exceeding even the increased limit for the alternative endpoint.

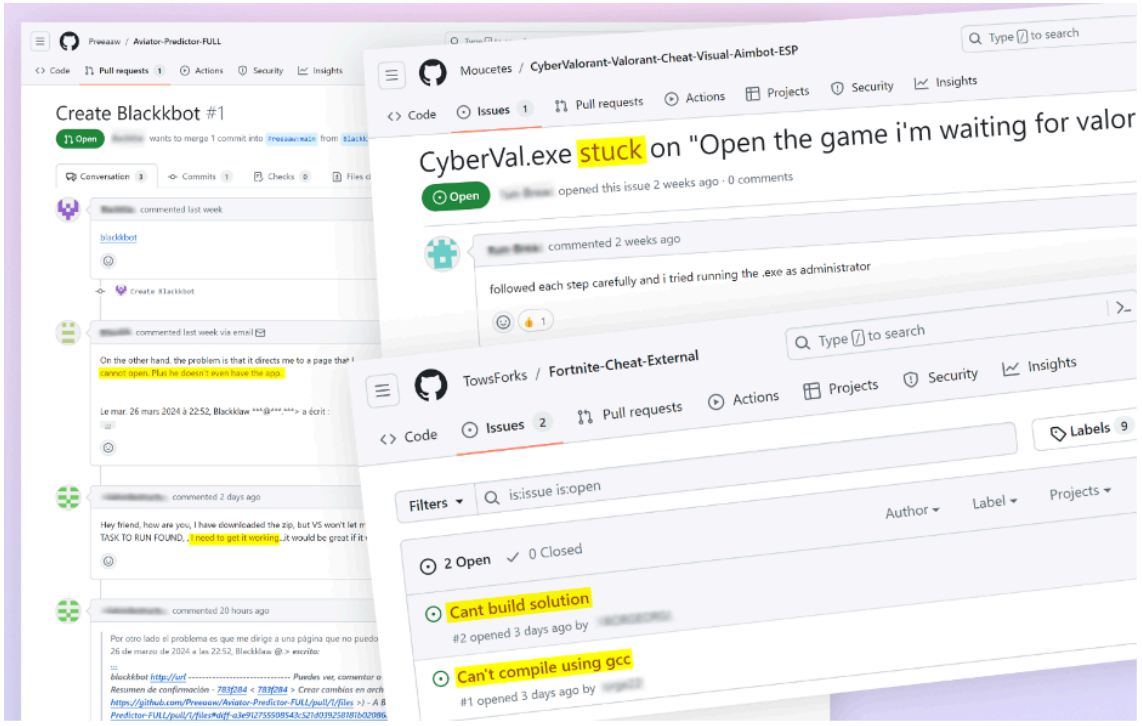


The results of our analysis of this malware suggest that the malware contains similarities to the “Keyzetsu clipper” malware, a relatively new addition to the growing list of crypto wallet clippers commonly distributed through pirated software.

This executable file also attempts to create persistence on Windows machines. It achieves this by creating a shortcut to the exe file and then establishing a daily scheduled task named “Feedback_API_VS_Services_Client” that executes the shortcut at 4AM. Notably, this task is created without any confirmation prompts, making it stealthier and more likely to go unnoticed by unsuspecting users.

Indicators of Successful Exploitation

Evidence indicates that the attackers’ campaign has successfully deceived unsuspecting users. Numerous malicious repositories have received complaints through Issues and pull requests from users who experienced problems after downloading and using the code.



Conclusion

The use of malicious GitHub repositories to distribute malware is an ongoing trend that poses a significant threat to the open-source ecosystem. By exploiting GitHub’s search functionality and manipulating repository properties, attackers can lure unsuspecting users into downloading and executing malicious code.

To prevent falling victim to similar attacks, it is recommended to keep an eye on the following suspicious properties of a repo:

1. Commit frequency: Does the repo have an extraordinary number of commits relative to its age? Are these commits changing the same file with very minor changes?
2. Stargazers: Who is starring this repo? Do most of the stargazers appear to have had accounts created around the same time?

By being aware of these red flags, users can better protect themselves from inadvertently downloading and executing malware.

In the aftermath of the XZ attack and many other recent incidents, it would be irresponsible for developers to rely solely on reputation as a metric when using open-source code. A developer who blindly takes code also blindly takes responsibility for that code. These incidents highlight the necessity for manual code reviews or the use of specialized tools that perform thorough code inspections for malware. Merely checking for known vulnerabilities is insufficient.

As part of Checkmarx's commitment to supply chain security, our research team continuously monitors and detects suspicious activities in the open-source software ecosystem. We track and flag potential indicators of malicious behavior and promptly alert our customers and the community to help protect them from these evolving threats.

Working together to keep the open-source ecosystem safe.

IOC

- hxxps[:]//cdn.discordapp[.]com/attachments/1192526919577649306/1211404800575537304/VisualStudioEN.7z?ex=6612fda3&is=660088a3&hm=5ae3b1b5d2c7dc91a9c07a65dbf8c61d3822b1f16a2d7c70eb37a039979e8290&
- hxxps[:]//cdn.discordapp[.]com/attachments/1192526919577649306/1211403074799804476/VisualStudioRU.7z?ex=6612fc07&is=66008707&hm=0a7fc9432f5ef58960b1f9a215c3feceb4e7704afd7179753faa93438d7e8f54&
- 08b799d56265e93f6aae4f089808d1cb
- cc9d54b78688ef6f41e4f4d0c8bced3e04bfcedc
- ooocyber[.]keenetic[.]pro
- 188[.]113[.]132[.]109
- https://rentry.co/MuckCompanyMMC/raw
- hxxps[:]//rentry[.]co/hwqfx/raw
- hxxps[:]//rentry[.]co/q3i7zp/raw
- hxxps[:]//rentry[.]co/tvfwh/raw
- hxxps[:]//cdn[.]discordapp.com/attachments/1193658583947149322/1218876343232630844/main.exe?ex=6609420d&is=65f6cd0d&hm=f5a0af7499e892637935c3e4071f2dc59d48214f56a1c1d7aedc3392f58176db&
- hxxps[:]//paste[.]fo/raw/dd6cd76eb5a0
- hxxps[:]//paste[.]fo/raw/efda79f59c55
- hxxps[:]//rentry[.]co/4543t/raw
- hxxps[:]//rentry[.]co/a2edp
- hxxps[:]//textbin[.]net/raw/gr2vzmwcvt

Tags:

Breaking News

Checkmarx Security Research Team

English

Leadership

Open-Source Security

supply chain attack

Supply Chain Security

Source: https://checkmarx.com/blog/new-technique-to-trick-developers-detected-in-an-open-source-supply-chain-attack/