AI Assisted Fake GitHub Repositories Fuel SmartLoader and LummaStealer Distribution

Published: 2025-03-11 · Archived: 2026-04-02 12:06:20 UTC

- Trend Research uncovered a campaign that uses fake GitHub repositories to distribute SmartLoader, which is then used to deliver Lumma Stealer and other malicious payloads. These repositories disguise malware as gaming cheats, cracked software, and system tools to deceive users.
- The campaign leverages GitHub's trusted reputation to evade detection, using AI-generated content to make fake repositories appear legitimate. Malicious ZIP files contain obfuscated Lua scripts that execute harmful payloads upon extraction.
- If the attack succeeds, threat actors can steal sensitive information like cryptocurrency wallets, two-factor authentication (2FA) extensions, login credentials, and other personally identifiable information (PII) that can potentially lead to identity theft and financial fraud.
- Cybercriminals are adapting from using GitHub file attachments to creating entire repositories, incorporating social engineering tactics and AI-assisted deception.
- Organizations and individuals should adopt proactive best practices, such as downloading software only from official sources, verifying repository authenticity, enabling security tools, and educating users on social engineering risks to mitigate such threats.

Cybercriminals are using fake GitHub repositories that make heavy use of AI for its lures to distribute malware, deceiving users with seemingly legitimate tools while evading detection. The Trend Micro Threat Hunting team identified an ongoing campaign that uses these repositories to deploy SmartLoader, which is then subsequently used to deliver other malware such as Lumma Stealer, an information stealer being distributed via the Malware-as-a-Service (MaaS) model by its creators (which we track as Water Kurita). These malicious repositories are disguised as non-malicious tools, including game cheats, cracked software, and cryptocurrency utilities. The campaign entices victims with promises of free or illicit unauthorized functionality, prompting them to download ZIP files (e.g., Release.zip, Software.zip). Upon execution, these files deploy SmartLoader, which ultimately facilitates the delivery of Lumma Stealer.

The campaign exploits GitHub's trusted reputation to evade initial detection, using social engineering tactics to target users searching for gaming mods, software cracks, or other tools. This operation highlights how legitimate platforms can be abused for malware distribution, emphasizing the dangers of downloading unverified "gray-area" tools, even from seemingly legitimate sources.

Abusing GitHub for malicious activities

The use of GitHub for malware distribution is not a new tactic, but it remains a persistent threat due to limited detection capabilities. While earlier campaigns primarily leveraged GitHub for file hosting, threat actors have since evolved their approach. They are now exploiting GitHub's trusted reputation more aggressively by using generative AI to create convincing fake repositories. As cybercriminals continue to innovate, this strategy is expected to expand, further reinforcing GitHub's role as a key vector for malware delivery.

One example of malware propagation via fake GitHub repositories can be seen on X/Twitter. Security researchers frequently tweet about open-source tools that are hosted on GitHub, and threat actors may exploit this by pretending to be researchers themselves. This tactic is designed to lure unsuspecting users into downloading malicious and/or fake tools under the guise of legitimacy.

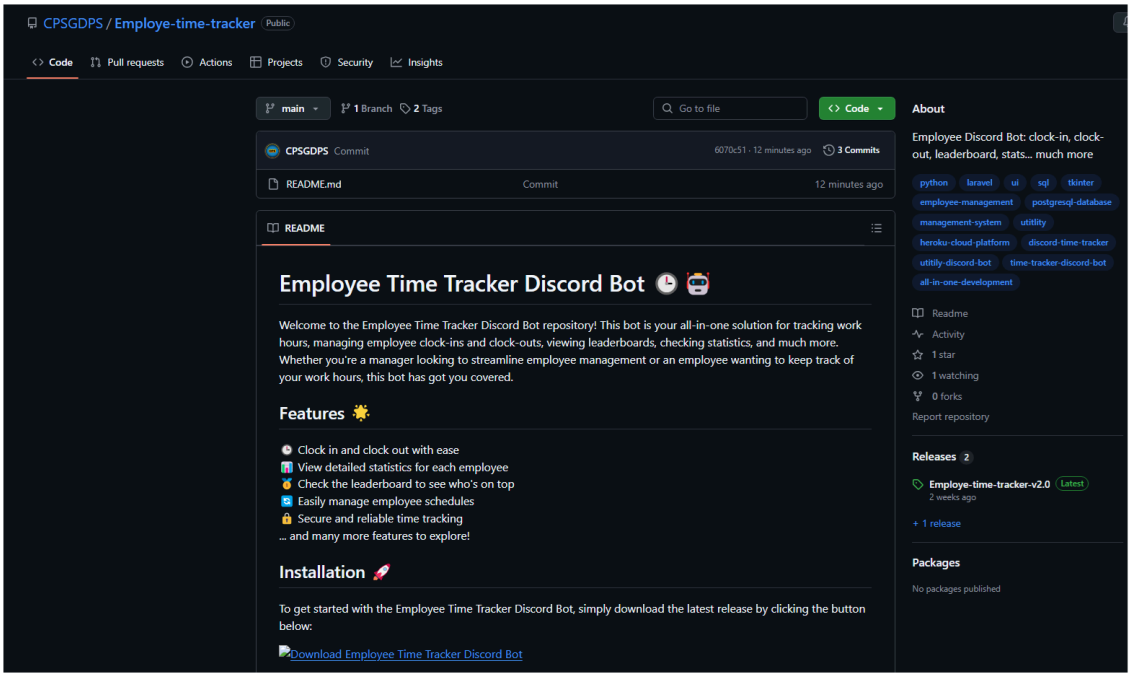
Additionally, SmartLoader is typically delivered via ZIP files containing obfuscated scripts. The only malicious component within the archive is an obfuscated Lua script, while the remaining files appear benign.

Technical analysis

Initial lure

The malware campaign begins with a fake GitHub repository designed to appear legitimate, often mimicking popular tools or software. By exploiting GitHub's trusted reputation, attackers deceive users into downloading malicious files. The malicious actors use AI to generate convincing README files and documentation, making these repositories appear professional and authentic — therefore increasing the likelihood that users will trust and install the malicious content.

The fake repositories stand out due to the formatting of its README content, which appears to be heavily assisted by AI, as exhibited by telltale signs such as excessive emoji usage, unnatural phrasing, a hyperlinked logo, and structured content. The repository contains only a README file, and all the hyperlinks redirect to the malicious files that are strategically concealed in the releases section, making them less conspicuous to unsuspecting users.



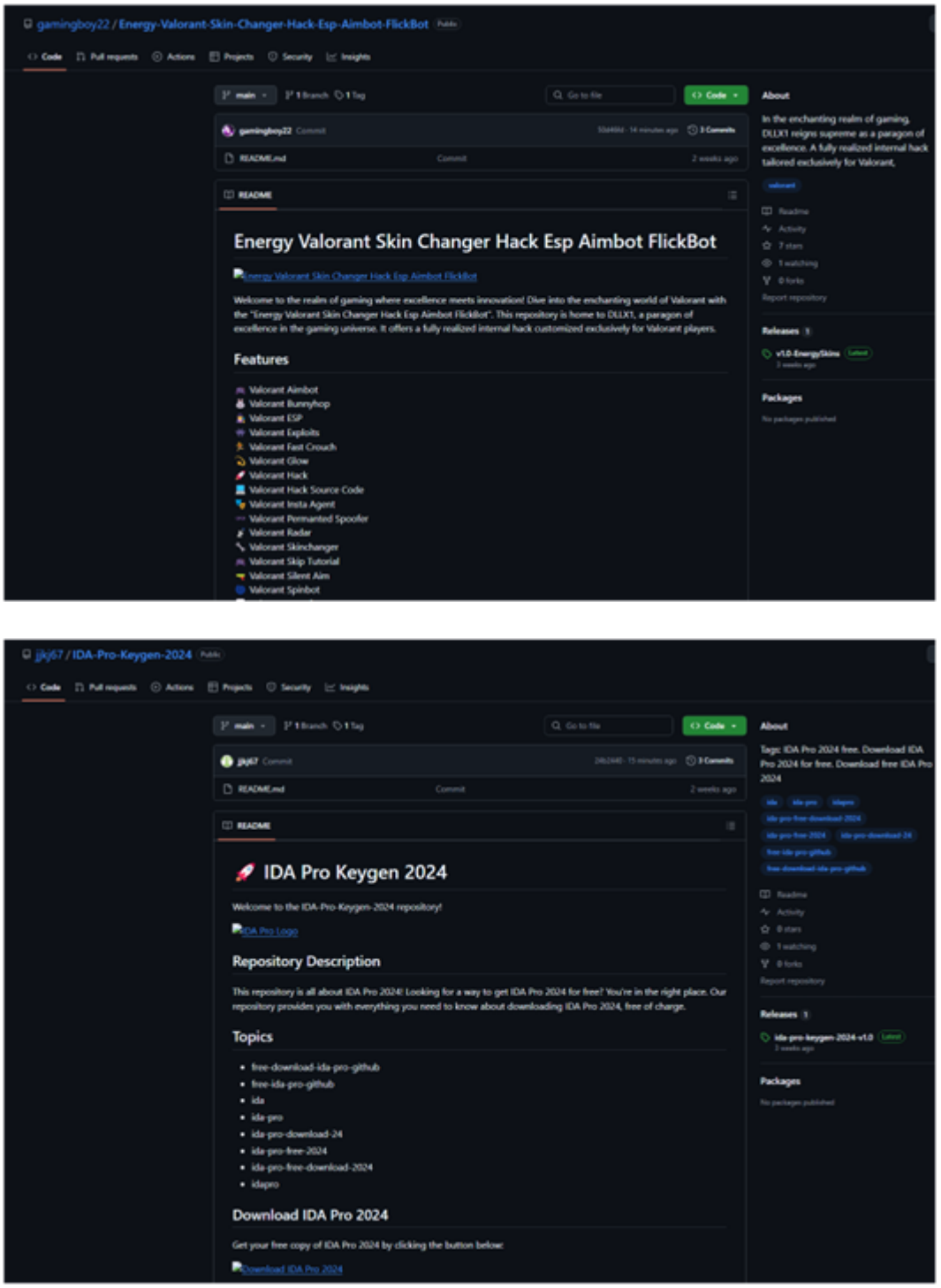


Figure 1. Examples of fake GitHub repositories

The primary goal of these repositories is to lure users into unknowingly downloading ZIP files that contain the SmartLoader payload. Once extracted, the ZIP file contains four components:

- lua51.dll: the LUAJIT runtime interpreter
- luajit.exe: the Lua loader executable
- userdata.txt: a malicious Lua script
- **Launcher.bat**: a batch file used to execute luajit.exe with the "userdata.txt" passed as an argument

While the executable and DLL files themselves are not malicious, the batch file is specifically designed to launch *luajit.exe*, which in turn executes the Lua script concealed within *userdata.txt*. This script serves as the true malicious payload, responsible for compromising the victim's system.

Similarities and differences with previous SmartLoader campaigns

In October 2024, a blog entryopen on a new tab was published analyzing the technique of using malicious Lua scripts alongside other malware components. The research provides a detailed breakdown of the Lua script, including its deobfuscation process and expected behavior.

Below is a summary of its key findings:

- The malware is distributed through obfuscated Lua scripts embedded in ZIP archives.
- A batch file triggers *Compiler.exe*, which loads *lua51.dll* and executes the malicious script.
- The loader connects to a command-and-control (C&C) server to receive and execute tasks.
- It uses *Prometheus Obfuscator* and the *ffi* library to hinder analysis and protect code integrity.
- It establishes persistence via scheduled tasks while collecting system information.
- It executes commands to evade security defenses, download payloads, and maintain persistence.
- It deploys *CypherIT Loader/Crypter* and *Redline*, the latter of which is a well-known infostealer with a thriving dark web market for selling stolen credentials.

Additionally, another reportopen on a new tab details a related campaign that follows the same strategy, but with Lumma Stealer as the payload. Both campaigns employ the same techniques and are delivered via SmartLoader.

Similarities with the October 2024 campaign (fake webpages) include the following:

- **Delivery infrastructure:** Malicious files were hosted on GitHub under user/file-attachments.
- **Lure mechanism:** Fake webpages mimicked legitimate software download sites. Clicking the "Download" button triggered malware retrieval from GitHub.

Meanwhile, the present campaign involving fake GitHub repositories has evolved in the following ways:

- **Shift in hosting strategy:** Instead of using GitHub file attachments, threat actors now store malicious files in the Releases section of fake repositories.
- **New lure mechanism:** AI-generated repository README files replaced fake webpages as the primary lure.
- **Evolving evasion tactics**: After the October 2024 campaign was uncovered, threat actors quickly evolved their tactics to evade detection while continuing to abuse GitHub's trusted status.

The operators behind SmartLoader demonstrated their adaptability by shifting from GitHub file attachments to repositories while maintaining their core techniques—such as obfuscated Lua scripts and batch execution chains. This highlights their focus on operational resilience despite growing security scrutiny.

SmartLoader to Lumma Stealer

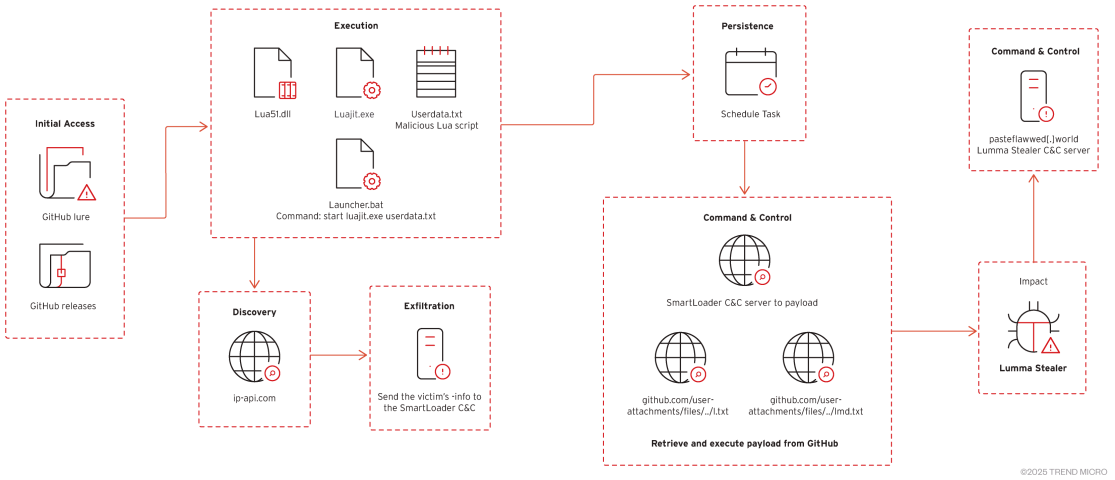


Figure 2. Attack chain

Figure 2 illustrates the complete attack chain—from the initial lure all the way to payload delivery. It details each stage of the process, highlighting the execution flow from SmartLoader to the LummaStealer payload.

Meanwhile, Figure 3 complements this by displaying the corresponding network traffic, captured in Wireshark, from the moment Smart Loader executes until the LummaStealer payload is delivered.

No.	Time	Source	Destination	Protocol	Lengtl	Info
6	0.059167		208.95.112.1	HTTP	226	GET /json/ HTTP/1.1
8	0.340539	208.95.112.1		HTTP/JSON	554	HTTP/1.1 200 OK , JSON (application/json)
4434	2.912991		213.176.73.80	SMARTLOADER	6347	PUT /api/YTAsODYsODIsOWQsYTEsODgsOTAsOTUsNjUsN2Qs HTTP/1.1 , JSON (application/json)
4494	5.237701	213.176.73.80		SMARTLOADER	1002	HTTP/1.1 200 OK , JSON (application/json)
4501	5.355199		20.205.243.166	HTTP	260	GET /user-attachments/files/18865448/lmd.txt HTTP/1.1
4503	5.394927	20.205.243.166		HTTP	177	HTTP/1.1 301 Moved Permanently
9146	10.427051		213.176.73.80	SMARTLOADER	441	PUT /task/YTAsODYsODIsOWQsYTEsODgsOTAsOTUsNjUsN2Qs HTTP/1.1 , JSON (application/json)
9148	10.836492	213.176.73.80		SMARTLOADER	787	HTTP/1.1 204 No Content
9150	10.837290		20.205.243.166	HTTP	258	GET /user-attachments/files/18722245/l.txt HTTP/1.1
9152	10.887839	20.205.243.166		HTTP	175	HTTP/1.1 301 Moved Permanently
10142	12.432242		213.176.73.80	SMARTLOADER	441	PUT /task/YTAsODYsODIsOWQsYTEsODgsOTAsOTUsNjUsN2Qs HTTP/1.1 , JSON (application/json)
10146	12.824969	213.176.73.80		SMARTLOADER	785	HTTP/1.1 204 No Content
10159	37.046749			DNS	78	Standard query 0x51cc A pasteflawwed.world
10160	37.058575			DNS	146	Standard query response 0x51cc No such name A pasteflawwed.world SOA v0n0.nic.world

Figure 3. Malicious packets for SmartLoader and Lumma Stealer

During its routine execution, the loader delivers the LummaStealer malware as its final payload. It retrieves a file from GitHub, saving it as *search.exe***,** which then executes LummaStealer. This executable drops the necessary files and runs an encrypted *AutoIt* script hidden within Excel files.

SmartLoader is capable of inflating files upon loading the payload, increasing the file size to approximately 1GB. The **IOC section** also includes the hashes for the encoded text files (*l.txt* and *lmd.txt*) that are downloaded from the GitHub links below.

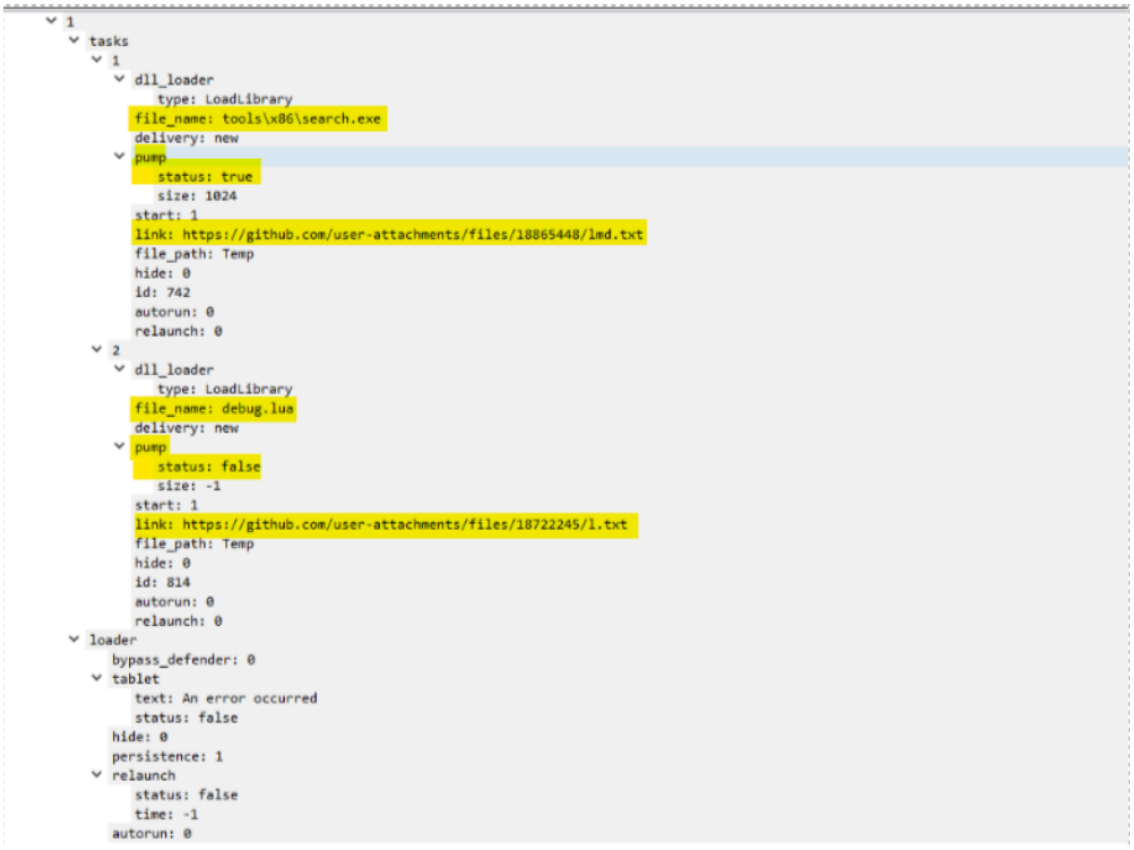


Figure 4. SmartLoader retrieves the files "lmd.txt" and "l.txt" from GitHub, renaming them to "search.exe" and "debug.lua"

After retrieving and executing the downloaded files, it will initiate a series of commands that create and execute additional files, setting the stage for the LummaStealer payload. It will perform malicious behaviors such as creating of malicious files in %TEMP% folder, concatenating into one batch script, and executing it:

cmd /c copy /bc..\Entertaining.xls + ..\Divide.xls + ..\Providence.xls + ..\Shakespeare.xls + ..\Adolescent.xls + ..\Divided.xls + ..\Unnecessary.xls + ..\Karma.xls

It will also perform multiple security software discovery commands via *findstr* such as:

findstr /I "opssvc wrsa"
findstr "AvastUI AVGUI bdservicehost nsWscSvc ekrn SophosHealth"

From the batch script, it will create another file named “Research.com” in the %TEMP% folder which is a misnamed AutoIT interpreter. Lastly, it will perform browser debugging using this command before reaching out to its C&C server:

"C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" --profile-directory="Default" --remote-debugging-port=9222

Finally, LummaStealer reaches out to its C&C) server at pasteflawwed[.]world. This communication channel is used to exfiltrate logs and other sensitive information harvested from the infected system.

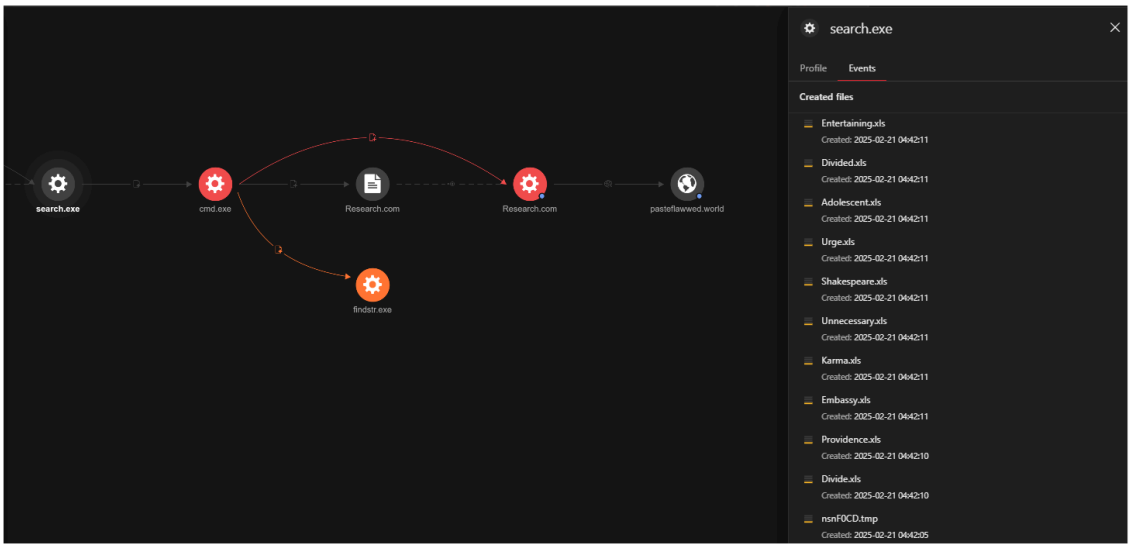


Figure 5. LummaStealer execution using misnamed AutoIt interpreter and eventually connecting to its C&C domain as seen in Trend Vision One™

Cybercriminals can use malware delivered via GitHub to perform highly destructive attacksopen on a new tab, especially when combined with advanced threats such as Lumma Stealer, which can gather information from web browsers, compromise cryptocurrency wallets and 2FA extensions, and steal sensitive data such as login credentials, financial information, and other PII. This can leave victims vulnerable to identity theft, financial fraud, and unauthorized access to critical accounts, resulting in severe financial and personal consequences. Furthermore, threat actors can exploit this stolen data even further by selling it to other cybercriminals for profit, further amplifying the risks to victims.

These attacks highlight how AI-driven cyber threats and sophisticated malware like Lumma Stealer are lowering the barrier for hackers to compromise both personal and professional accounts. As cybercriminals increasingly make use of advanced tools to automate and enhance their attacks, the urgency for stronger cybersecurity measures becomes clear. Implementing robust defenses is crucial to mitigating these rapidly evolving threats.

Mitigation and recommendations

To defend against threats like **SmartLoader** and similar malware campaigns, individuals and organizations should consider the following best practices:

- **Download software only from official sources:** Avoid third-party sites, torrents, and repositories that offer free or cracked software.
- **Verify repository authenticity:** Check for legitimate contributors, repository history, and signs of AI-generated or suspicious documentation.
- **Enable security features:** Use endpoint security solutions that detect and block malicious downloads.
- **Analyze files before execution:** Use sandboxing tools to scan unknown files before running them.
- **Implement network security controls:** Block known malicious GitHub repositories and restrict file downloads from unverified sources.
- **Monitor for abnormal activity:** Use security information and event management tools to detect unauthorized script executions and unusual outbound connections.

- **Educate employees on social engineering risks:** Conduct security awareness training to prevent employees from falling for fake repositories.
- **Enforce application control policies:** Apply measures to prevent execution of unauthorized applications and scripts.

By following these best practices, both users and enterprises can reduce the risk of falling victim to malware campaigns that exploit trusted platforms like GitHub. Cybercriminals will continue to adapt, but a proactive security approach will help mitigate these evolving threats.

Proactive security with Trend Vision One™

Trend Vision One™ one-platformis an enterprise cybersecurity platform that simplifies security and helps enterprises detect and stop threats faster by consolidating multiple security capabilities, enabling greater command of the enterprise's attack surface, and providing complete visibility into its cyber risk posture. The cloud-based platform leverages AI and threat intelligence from 250 million sensors and 16 threat research centers around the globe to provide comprehensive risk insights, earlier threat detection, and automated risk and threat response options in a single solution.

Trend Micro™ Threat Intelligence

To stay ahead of evolving threats, Trend Vision One customers can access a range of Intelligence Reports and Trend Vision One™ Threat Insights. These help customers stay ahead of cyber threats before they happen and allows them to prepare for emerging threats by offering comprehensive information on threat actors, their malicious activities, and their techniques. By leveraging this intelligence, customers can take proactive steps to protect their environments, mitigate risks, and effectively respond to threats.

Trend Vision One Intelligence Reports App [IOC Sweeping]

- *From SmartLoader to LummaStealer: AI-Generated fake GitHub repositories delivering malware*

Threat Insights App

- **Threat Actors**: Water Kurita
- **Emerging Threats:** From SmartLoader to LummaStealer: AI-Generated fake GitHub repositories delivering malware

Hunting queries

Trend Vision One Search App

Trend Vision One customers can use the Search App to match or hunt the malicious indicators mentioned in this blog post using data within their environment.

LummaStealer connection to C&C server

eventSubId:301 AND processFilePath:Research.com AND hostName:pasteflawwed.world

More hunting queries are available for Trend Vision One customers with Threat Insights entitlement enabledproducts.

Conclusion

The ongoing campaign using fake GitHub repositories to distribute SmartLoader and Lumma Stealer highlights the evolving tactics of cybercriminals. By abusing GitHub's trusted reputation, attackers can use social engineering techniques and AI-generated content to lure victims into downloading malicious files. The shift from traditional GitHub file attachments to full repositories demonstrates their adaptability in evading detection and maintaining operational resilience.

As cyber threats continue to evolve, organizations and individual users must remain vigilant against such deceptive tactics. This campaign underscores the importance of verifying software sources, especially when dealing with open-source platforms.

Indicators of compromise

The indicators of compromise for this entry can be found here.

Tags
