

GitHub confirms compromised Nx Console extension was initial access vector

 threatlocker.com/blog/github-breach-likely-caused-by-nx-console-compromise

May 21, 2026

May 21, 2026

Written by:

ThreatLocker Threat Intelligence

This article has been updated to include announcements from [Jeff Cross, CEO of Nx](#), confirming Nx Console's role in the GitHub breach.

Malicious VS code extension causes GitHub breach

On May 19th, 2026, 11:48 p.m. UTC GitHub [announced on X](#) that they were investigating unauthorized access to their internal repositories. This announcement came after they detected an employee device utilizing a malicious VS code extension that had been poisoned.

Shortly after the public disclosure, TeamPCP announced that they will be auctioning off GitHub's source code to the highest bidder. Minor edits to the original post have been captured, with Lapsus being authorized to broker the sale of GitHub's data.

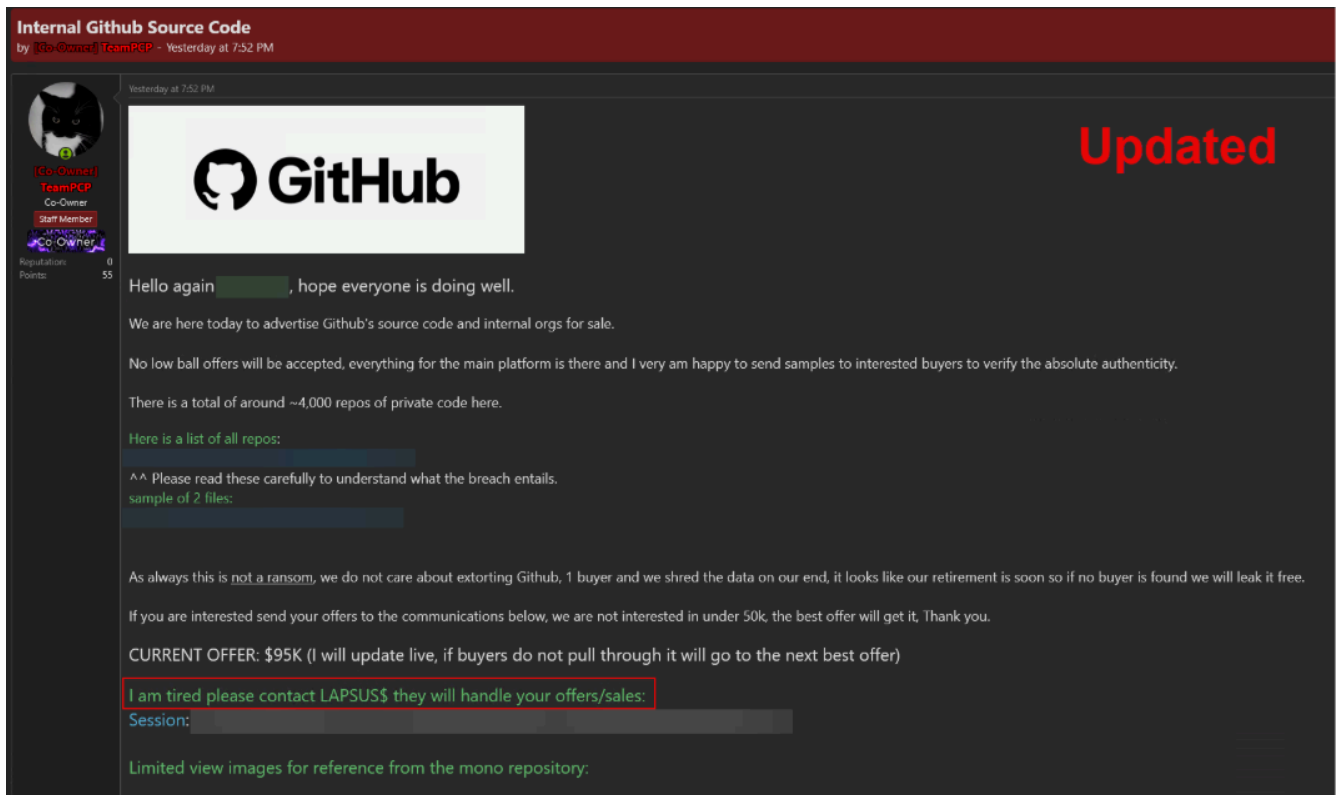


Figure 1: TeamPCP Data Auction

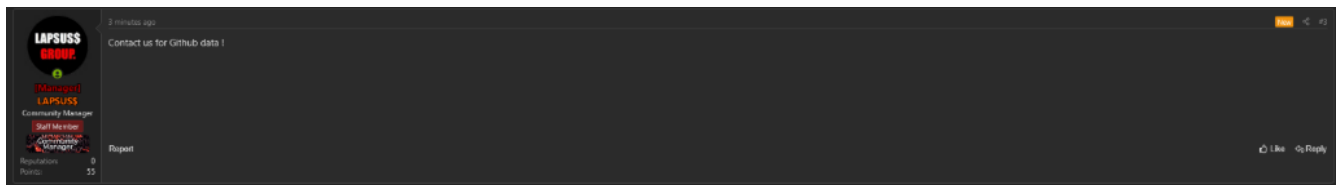


Figure 2: LAPSUS\$ Group

Prior to these events, on May 18, 2026, the compromised Nx Console version 18.95.0 was published. Our analysis has uncovered a shared artifact that was found present with the [recent @antv compromise](#) that was propagating Mini Shai-Hulud.

The malicious Nx Console extension and @antv packages created Python C2 backdoors at `~/.local/share/kitty/cat.py`. With the recent disclosures, timeline of events, and incident artifacts, [Nx Tools confirmed](#) that the GitHub compromise was a downstream effect of the malicious Nx Console packages.

Initial compromise stems from stolen GitHub credential

The initial compromise of the Nx Console extension stems from a stolen GitHub credential belonging to an Nx developer, allowing for the publishing and distribution of the backdoored version v18.95.0 of Nx Console. The affected extension bundle included a block of minified

and highly obfuscated JavaScript, `main.js`, which was designed to impersonate legitimate MCP server functionality and upon workspace activation would fetch a package titled “nx-next” from an orphaned commit in the official “nrwl/nx” repository.

Figure 3: main.js Injected Commands

This retrieved package JSON contained a `bin` entry pointing to the “`index.js`” file within the package, which serves as the second stage payload, also containing minified and highly obfuscated JavaScript with multiple layers.

Figure 4: package.json

Obfuscation methods used include a shuffled string table, PBKDF2-runtime string decryption, mangled and split identifiers, and encrypted blobs of additional data. To prevent detonation in analysis environments, the script additionally queries host CPU count and checks for a generated lock file within a temporary directory. Geolocation checks based on time zone and locale are invoked to avoid execution in CIS countries, including Russia.

Figure 5: Encrypted Strings

For persistence, a `kitty-monitor` daemon is installed along with `~/.local/share/kitty/cat.py` to provide a C2 backdoor into the environment. This simple C2 client queries GitHub APIs for a commit containing the keyword “`firedalazer`” and parses the response for two blocks of base64 encoded data: a URL and a signature. If the signature is verified with the URL and an embedded public key, a Python script is captured from the given URL, written to a random file in `/tmp`, and executed locally. Although incredibly simple, the fact that this client depends on RSA signature verification means that only messages sent by the attacker would be accepted and executed.

Figure 6: cat.py C2 Installation

Figure 7: Github Query

Figure 8: download_and_execute Function

Several secrets and credentials are then searched for to extract and exfiltrate including PyPi, NPM, Bitwarden, 1Password (OP), AWS, and many others. AWS tokens and secrets are captured by querying the APIPA address “169.254.169.254”, which is used to host local AWS IMDS instances.

Figure 9: Captured Tokens and Secrets

Figure 10: AWS IMDS Instance

The /proc path on linux allows for processes to query process information which includes process memory and its contents. The Github Actions Runner .Worker process has been known to store tokens, credentials, and other plaintext secrets in memory. The findRunnerWorkerPIDLinux function appears to find and query this process for any valuable data to extract and exfiltrate.

Figure 11: Github Runner.Worker Process

As seen by previous implementations of [Mini Shai Hulud](#), functionality is added at the end of the script to delete the home ~/ and ~/Documents folders of the victim if the dead drop repositories are deleted before persistence is removed.

Figure 12: Delete Home and Documents

Mitigations

TeamPCP targets developer credentials by deploying info-stealing malware through compromised supply chains. By following best practices for cloud and web application, security organizations can mitigate the risk(s) posed by these threat actors:

Proper credential management and storage, ensuring that credentials are not saved as clear text.

Service accounts should be configured to require MFA and not just an API key.

Enforce IMDSV2 on all AWS services and allowlisting from approved S3 resources.

Implement authentication with least privilege principles to Docker and Kubernetes services.

IOCs

Malicious Extension and Package Indicators

- nrwl.angular-console
- Malicious version 18.95.0
- nxConsole.mcpExtensionInstalledSha
- VS Code globalState value
558B09D7AD0D1660E2A0FB8A06DA81A6F42E06D2
- github:nrwl/nx#558b09d7

File Hashes

- Malicious VSIX, Nx Console v18.95.0
SHA-256:
1A4AFCE34918BDC74AE3F31EDAFFFFAA0EE074D83618F53EDFD88137927340B8
- Malicious main.js inside VSIX
SHA-256:
B0CEFB66B953E5184B6ADB3035E9E267335AC5EABFE1848E07834777B9397B74
- Obfuscated payload index.js from orphan commit
SHA-256:
E7347D90653EFC565F03733A95E9209D78F9CFA81E31FF2B2DD9D48D75A4B8B1
- Dropper package.json
SHA-256:
43F2B001846C4966073EBFFA5BE8F15E491A1E7D32BBD805D57406FF540E0DD9
- C2 Client cat.py
SHA-256:
FB5C97557230A27460FDAB01FAFCFABEAA49590BAFD5B6EF30501AA9E0A51142

Git Object Hashes

- Malicious orphan commit
SHA: 558B09D7AD0D1660E2A0FB8A06DA81A6F42E06D2
- Commit tree
SHA: BA642FE2C7C65E42DD7F6444B83023DC6827E08C
- index.js blob
SHA: ACFC3F957A63B4CDE93FF645F2B6BF26A8ED1BBF
- package.json blob
SHA: 9D88F040C44B5F4D5F9DB15FF89310776C168E99

Persistence and Filesystem Artifacts

- ~/.local/share/kitty/cat.py
- ~/Library/LaunchAgents/com.user.kitty-monitor.plist
- /var/tmp/.gh_update_state
- /tmp/kitty-*
- path containing nrwl.angular-console-18.95.0
- entries referencing github:nrwl/nx#558b09d7

Process Indicators

- python or python3 process running cat.py
- Any process with environment variable __DAEMONIZED=1
- bun, node, or related process executing from /tmp/kitty-*
- VS Code-spawned process executing GitHub/Nx payload logic

Network Indicators

- api.github.com/search/commits?q=firedalazer
- api.github.com/search/commits?q=firedalazer&sort=committer-date&order=desc&per_page=1
- bun.sh/install
- DNS tunneling queries
- Unexpected outbound HTTPS traffic on port 443 from developer machines
- Encrypted C2 over port 443
- 169.254.169.254 (AWS IMDS instance)
- 169.254.170.2 (AWS ECS instance)
- 127.0.0.1:8200
- [fulcio\[.\]sigstore\[.\]dev](#)
- [rekor\[.\]sigstore\[.\]dev](#)

RELATED blogS
