North Korean remote workers landing jobs in the West | ThreatLabz

By Seongsu Park

Published: 2024-11-04 · Archived: 2026-04-02 11:03:33 UTC

Technical Analysis

Contagious Interview campaign as initial attack vector

The initial infection method for the Contagious Interview campaign has been well-documented by the security industry and remains largely unchanged, so it will not be covered in detail here. ThreatLabz has since observed new Contagious Interview campaign attacks, where a threat actor posted a job opening for a full-stack developer on part-time hiring platforms, like Freelancer. As part of the interview process, applicants were asked to solve a coding problem on GitHub and submit their results. However, the GitHub repository, which is controlled by the attacker, contained malicious JavaScript code named “BeaverTail”. The figure below shows a fake job opportunity posted as part of the Contagious Interview attack.

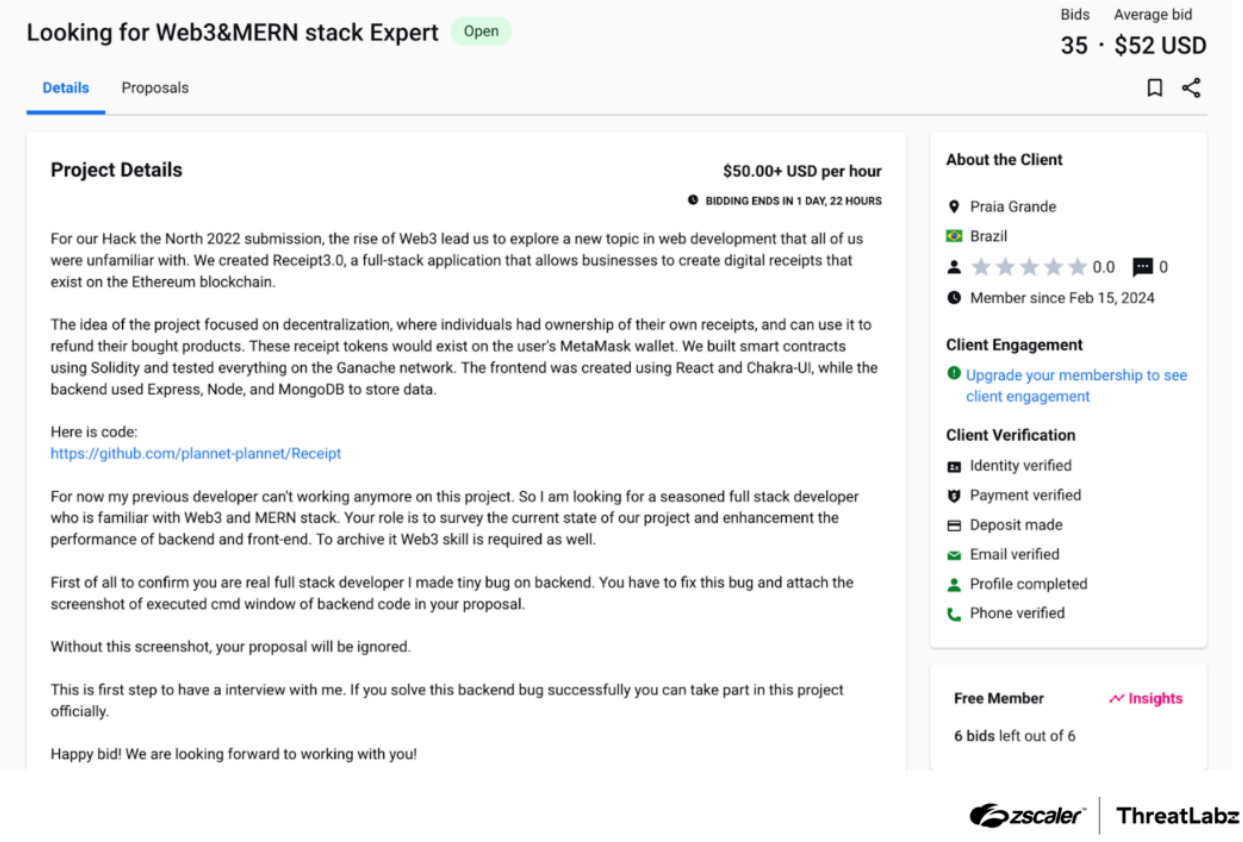

Figure 2: Fake job opening that delivers a malicious NPM package thus initiating an Contagious Interview campaign infection.

The threat actors aggressively contact potential victims through social media platforms, focusing on web, cryptocurrency, and AI developers. Additionally, the threat actors heavily rely on source code publishing platforms to host malicious files such as GitHub, GitLab, and BitBucket.

BeaverTail and InvisibleFerret infection chain

The JavaScript executed by the initially delivered package, BeaverTail, has undergone minimal changes since its initial discovery. For a long time, the threat actor used a malicious NPM package as an initial infection vector. However, while closely monitoring this campaign, we discovered they have adopted different file types to deliver the payload, like macOS applications and Windows Installers disguised as chat applications as shown in the figure below.

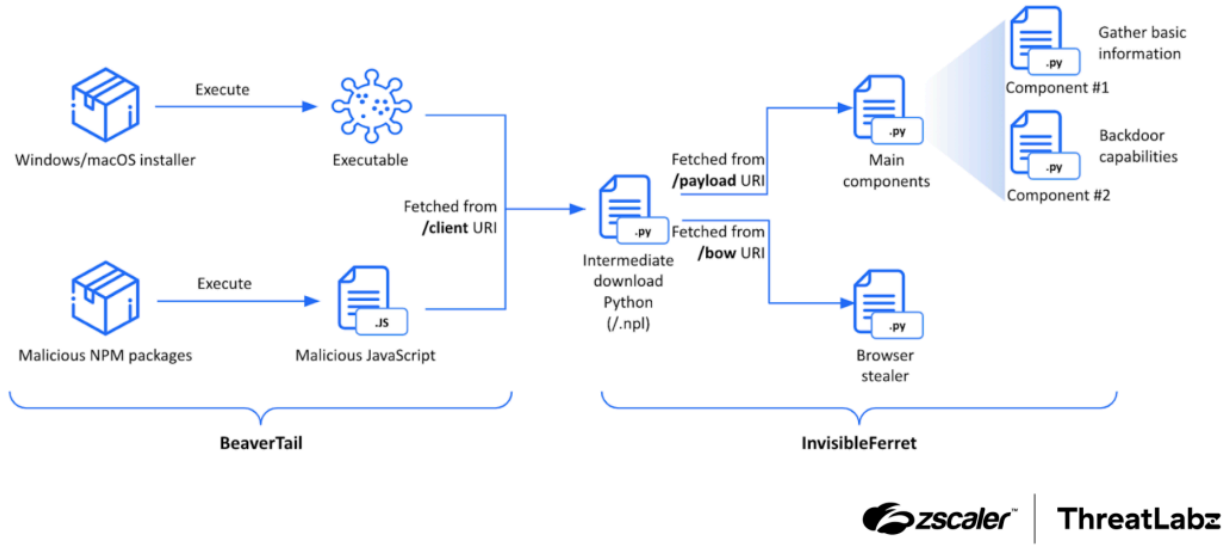

Figure 3: BeaverTail and InvisibleFerret infection chain

BeaverTail has adopted a new obfuscation technique to evade detection utilizing a JavaScript-obfuscator to mask its strings and functions. In some cases, additional malicious code is retrieved from attacker-controlled servers and dynamically executed by extracting the `cookie` property from the fetched JSON data and ran via the `eval` function. This highlights the effort the threat actor has put into further evading detection.

The Python script retrieved by BeaverTail can download additional Python scripts from the `/payload` and `/bow` URIs, including the main backdoor script and a script for stealing browser data. The main backdoor script, InvisibleFerret, has two components: sending basic system information and executing backdoor functionalities. The threat actor uses InvisibleFerret to exfiltrate data from victims, as shown in the figure below.

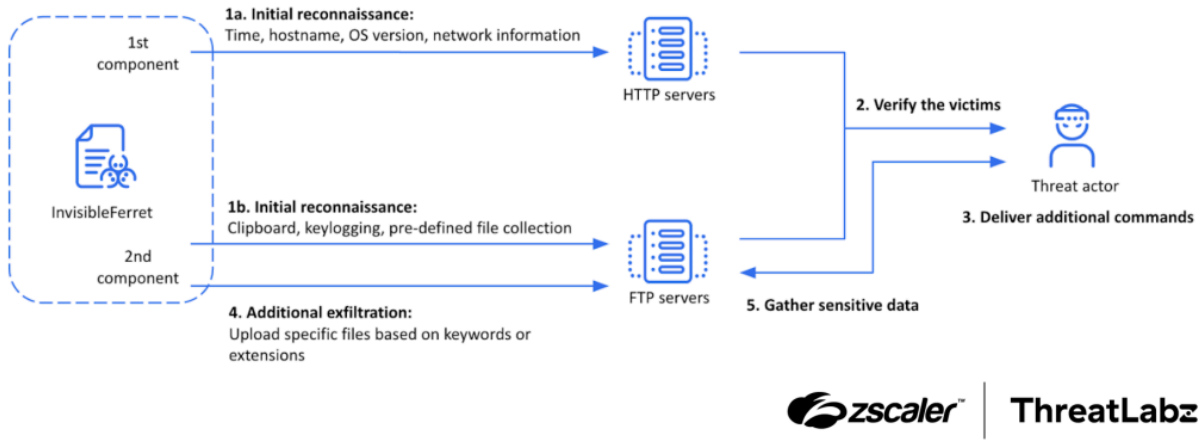

Figure 4: Contagious Interview campaign, which utilizes InvisibleFerret to exfiltrate data from a victim.

Upon execution, the InvisibleFerret script starts keylogging in a separate thread. The keylogging thread checks for changes in the active window, logs key presses, and captures clipboard content during copy and paste operations. After that, the threat actor usually delivers `ssh_clip` (sends stored keylogging data to the C2 server) or `ssh_env` (sends predefined sensitive data to the FTP server) commands to collect basic information from the victim and verify that the compromised host is valuable.

The threat actor may collect basic information about the victim using the aforementioned functionalities and begin to exfiltrate additional files if the victim is deemed a valuable target. For additional data collection, specific files are uploaded based on commands from the operators. The table below shows the commands supported by InvisibleFerret.

Commands	Description
`sdira`	Upload all files from a specified directory and its subdirectories.
`sdir`	Upload all files from a specified directory.
`sfile`	Upload a single file.
`sfinda`	Find and upload files matching a pattern in a directory and its subdirectories.
`sfindr`	Find and upload files matching a pattern in a directory (non-recursive).

Commands	Description
`sfind`	Find and upload files matching a pattern in the current directory and its subdirectories.

Table 1: InvisibleFerret backdoor commands used to exfiltrate files from a victim.

Using these file upload capabilities, the threat actor typically exfiltrates PDF documents, image files, and source code. Source code is often a target, because developers sometimes store credentials such as login IDs and passwords in plain text. Furthermore, by exfiltrating source code from victims, mainly those associated with cryptocurrency or web development, the threat actor can reuse the exfiltrated data for other campaigns, or potentially access and steal cryptocurrency.

In August 2024, the InvisibleFerret malware author added new backdoor commands, additional exfiltration targets, and communication channels. One new command internally called `ssh_zcp`, copies browser data like extensions and cryptocurrency wallet data. InvisibleFerret also copies application data directories for cryptocurrency wallets and password manager applications, targeting specific paths based on the operating system.

- **For Windows** (`.7z` format): Uses the `py7zr.SevenZipFile` library to compress and encrypt files with the provided password.
- **For non-Windows systems** (.zip format): Uses the `pyzipper.AESZipFile` library to create a ZIP archive with AES encryption, defaulting to the password `123`.

After creating the ZIP archive, InvisibleFerret sends the file to a Telegram chat using the provided token and chat ID. InvisibleFerret also uploads the ZIP archive to the specified FTP server with a `zdat_` prefix. The threat actor used FTP for exfiltration for an extended period. However, they have now removed the functionality for exfiltrating stolen data to an FTP server. Instead, InvisibleFerret now exclusively uses the HTTP protocol for file exfiltration via the `/uploads` URI. All of these changes suggest that InvisibleFerret is still under active development.

We recently discovered that the InvisibleFerret Python script has been modified. Now, its backdoor functionalities have been updated and heavily focused on executing an AnyDesk client (such as updating its password salt). Additionally, InvisibleFerret added a capability to create Startup scripts for different operating systems.

Commands	Description
`AA`	Collects cryptocurrency-related browser extensions and sends them to the C2 server through the `/uploads` URI.

Commands	Description
`AO`	Collects browser stored data and sends it to the C2 server through the `/uploads` URI.
`AB`	Collects configuration data from `service.conf` and `system.conf`, and sends it to the C2 server.
`Ab`	Checks if the `C:/Program Files (x86)/AnyDesk/AnyDesk.exe` file exists.
`AC`	Updates `pwd_hash`, `pwd_salt`, and `token_salt` configuration values for the AnyDesk client.
`AP`	Gathers system information and AnyDesk configuration files, then sends this data to the C2 server via the `/info` URI.
`AQ`	Gathers installed programs and running processes, and sends them to the C2 server via the `/data` URI.
`AR`	Extracts data from Microsoft Sticky Notes and sends that data to the C2 server via the `/data` URI.
`AD`	Downloads additional payloads from the `/bow` URI.
`n`	Set up a Startup script for different operating systems, such as Linux, Windows, and macOS. • Linux: Sets up a `.desktop` entry to run the script at Startup in GNOME-based Linux environments. • Windows: Creates a batch file (`queue.bat`) in the Startup folder to run a Python script. • macOS: Creates a `com.avatar.update.wake.plist` file to run the script on Startup via LaunchAgents.

Table 2: Commands supported by a newly discovered InvisibleFerret backdoor.

Distribution of operation systems infected by Contagious Interview

ThreatLabz has identified over 140 victims compromised by the Contagious Interview campaign within a two-month period. Interestingly, over half of these victims used Windows machines, while the other half used non-Windows systems, including Linux and macOS. This indicates that the campaign successfully compromised multiple platforms by leveraging OS-independent scripts such as JavaScript and Python. The figure below shows the distribution of victims' systems.

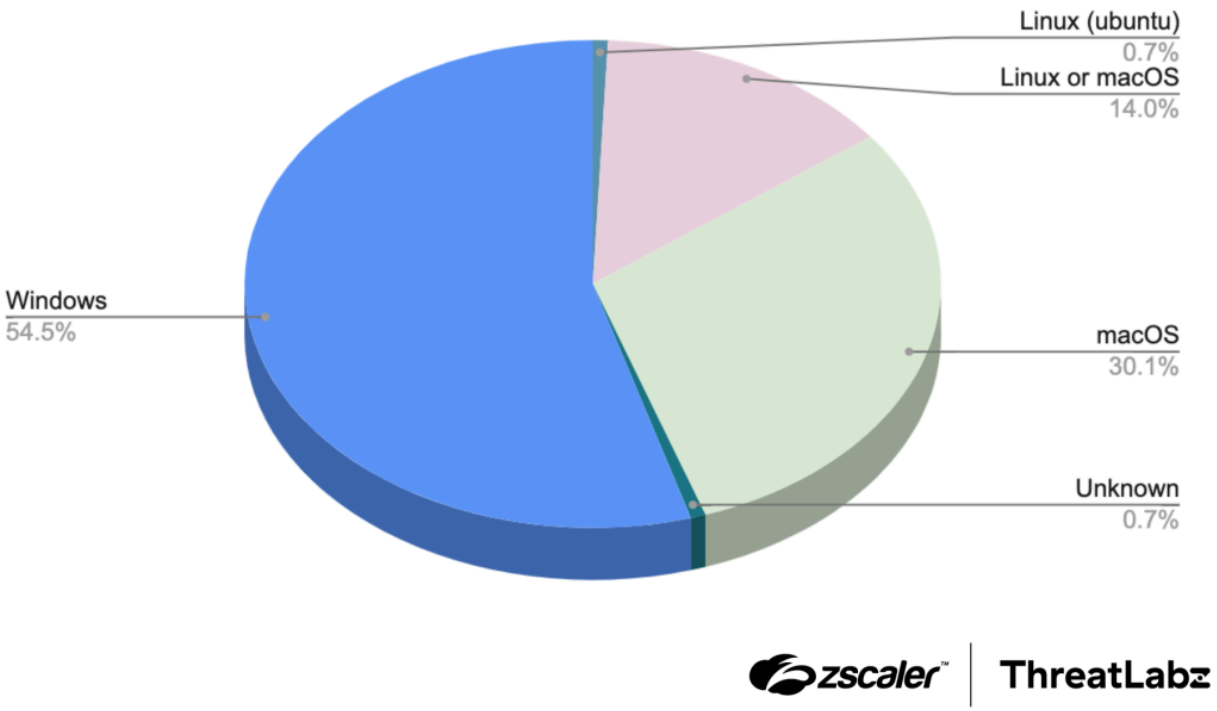

Figure 5: Distribution of operating systems infected by the Contagious Interview campaign.

The threat group exfiltrated cryptocurrency-related files from most of the victims. By targeting developers in the cryptocurrency industry, the threat group occasionally obtained files containing login credentials for critical systems. Furthermore, victims were not restricted to specific countries. Many of the victim developers were from India, Pakistan, Kenya, Nigeria, Spain, and Russia.

North Korean IT workers use WageMole to secure remote jobs in other countries

While monitoring the Contagious Interview campaign closely, we analyzed an associated campaign named WageMole being perpetrated by the same threat group. The WageMole campaign leverages a combination of social engineering and technology to secure legitimate remote job opportunities and earn money through their development skills. After a thorough investigation, we organized their operational process into several stages, all of which are shown in the figure below and discussed in detail.

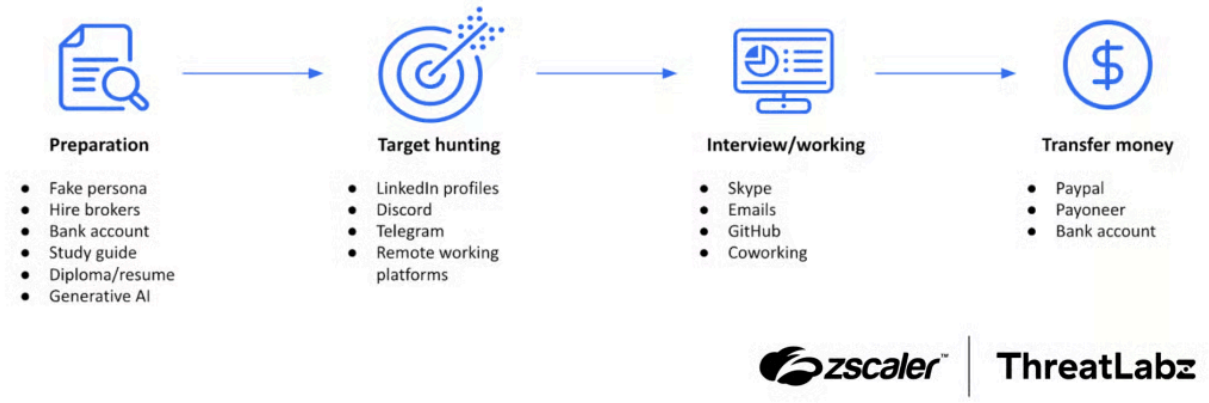

Figure 6: Operational process of WageMole campaign organized into stages.

Preparation

WageMole threat actors' first step in applying for a job involves creating fake personas. WageMole threat actors obtain fake passports or other forms of identification, either through the Contagious Interview campaign or by purchasing them from real individuals. Occasionally, they hire foreign nationals residing in the U.S. In addition, WageMole threat actors create fake driver's licenses to verify their identity. In these cases, they appear to use stolen driver's licenses, altering only the photo on the ID while leaving the rest of the information unchanged.

WageMole threat actors prepare study guides for the job interview process that include self introductions, work history, and answers to technical questions, as shown below.

- **Self introduction:** As a full-stack engineer, talk about Spring Boot, React/Next developer, Larvel, Symfony, Node.js, TypeScript, WordPress, ASP.NET, etc.
- **Working experience:** Describe teamwork experience, best and worst experience, a challenging project, development process, Agile/Scrum environment experience, difference between frontend and backend, Java Spring Boot developing experience, how to solve an issue, how to learn new technology, the reason to hire you, etc.
- **Technical questions:** Explain React.js, Flutter, Backend API development, and AI.
- **General questions:** Additional questions to ask employers for the hiring process and roles.

When WageMole threat actors created this study guide, we believe they used generative AI to derive the solutions to each question because:

- The answers are well-written and well-structured, and some of the answers start with "Certainly!".
- Most of the paragraphs are numbered and exhibit a formal style.
- When creating fake identity cards and passports, WageMole threat actors used an AI face editor to modify the person's photo. This included adding a smile, making the person look more professional, removing the background, and making the threat actor appear more Western.

WageMole threat actors create multiple versions of their resume for different roles, like full-stack or PHP developer, each listing different residency locations (e.g., U.S., U.K., Estonia). WageMole threat actors also collected publicly available certificate or diploma images from the internet to use in the interview process, often

from private education sites related to skills like ASP development, Android development, and machine learning. WageMole uses fake career histories, degrees, and alters details like company names and university affiliations, while keeping the same name and contact information once they create an identity.

Target hunting

These threat actors prefer LinkedIn for finding job opportunities. They create fake LinkedIn profiles, often portraying themselves as full-stack developers or AI engineers from several countries like Italy, Germany, Netherlands, Estonia, Switzerland, and Lithuania. We discovered several LinkedIn profiles used in this campaign, such as the following:

- hxxps://www.linkedin[.]com/in/frank-schoneberg-a089832a4/
- hxxps://www.linkedin[.]com/in/logan-collins-374404306
- hxxps://www.linkedin[.]com/in/adam-song05/

The figure below shows two of those fraudulent LinkedIn profiles.

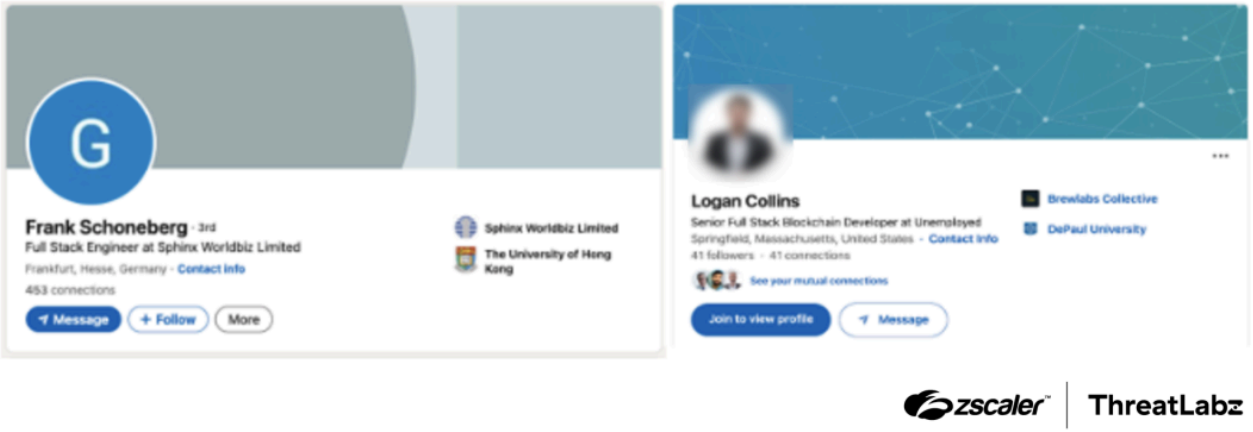

Figure 7: Fake LinkedIn WageMole profiles.

In addition to LinkedIn, the threat actors set up websites and GitHub repositories to showcase their skills and attract potential employers.

During the job search, WageMole threat actors aggressively use job seeking platforms such as Indeed, Glassdoor, Upwork, and cryptocurrency specialized sites such as degencryptojobs.com and web3.career. During the job hunting process, they search for remote roles like front/backend web developer, UX/UI designer, full-stack engineer, and blockchain developer. WageMole threat actors target various industries like: information technology, healthcare, retail, financial services, construction, and real estate. Several Fortune 500 companies, and even aerospace and defense companies, are included in WageMole's job search list. We can't confirm if WageMole threat actors wanted to be hired by the defense industry intentionally or if they were just searching for remote jobs and stumbled upon these positions. WageMole threat actors also prepared emails and message templates to send potential employers. In several of their messages to potential employers, the threat actor communicates in broken English, as shown below.

Sample 1
Hello,
I'm a senior Vue and Laravel developer with 8 years of experience in JS frameworks like MEAN/MEVN/MERN. I specialize in Vue, API integration, plugin customization, and bug fixing.
I'm ready to start your project, ensuring perfection in a short period. I take your project seriously, always striving for the best outcome and providing creative ideas when needed.
So let's talk and discuss.
Thanks!

Sample 2
As an accomplished web developer with a sharp eye for detail, your project fits right into my skillset.
The ability to work with and adjust existing text while maintaining consistency and tone is something I've consistently done throughout my career.
My expertise in HTML and CSS, combined with meticulous graphic design skills, will ensure that your website maintains a polished and professional appearance even after the adjustments are made.

Sample 3
Hi, Dear client. How are you?
I read your job post carefully and am excited about it.
As I am a senior full stack developer, I have over 7 years of experience in software development.
Especially, React, Node.js, React Native is my powerful skills.
I am sure that this job is appropriate to me greatly.
We can discuss the more detail via conversation.
I will wait to hear from you.
Have a nice day. Best regards.

Most of the templates are written in English, but we observed that WageMole threat actors also created Japanese versions, indicating potential interest in job opportunities in Japan.

When required, WageMole threat actors use automation scripts to create accounts on job search platforms, like Upwork.

In another instance, WageMole threat actors offered someone living in the U.S. $1,000 USD for access to their Upwork account and their computer.

Interview/working

WageMole threat actors use Skype to converse with a potential employer and during the interview process. Skype offers local phone numbers, including U.S. numbers and call forwarding, allowing remote workers to deceive employers about their location. Since larger companies often have more stringent background checks, WageMole threat actors typically target small to mid-sized businesses.

When a WageMole threat actor lacks the skills to answer interview questions, they often rely on a colleague with the necessary expertise to assist. During employment, WageMole actors collaborate with others within their threat

actor group by sharing code and solutions. WageMole threat actors also use GitHub to prepare for the hiring process and complete tasks. From their code paths (shown below), we can infer the services they provide.

- `D:\Work\Crypto\Crypo-backend\app`
- `D:\Work\Crypto\Crypo-frontend`
- `/home/Crypo-Telegram-Notification-Bot`
- `/home/frontend/components/page-parts/WalletsPage/HistoryPage`

Transfer money

The goal of WageMole is to generate funds using the threat group's professional skills. To bypass economic sanctions, WageMole needs secure methods to transfer money. In one case, a full-time employee earned an annual salary of 48,000 EUR, with monthly payments sent through a European bank. In another case, a remote worker earned 12 EUR per hour for 48 hours a week, totaling 550 EUR weekly. WageMole also frequently requests payments via online platforms like PayPal or Payoneer to evade monitoring and conceal their identity.

Explore more Zscaler blogs
