

GhostSocks - Lumma's Partner In Proxy

infrawatch.app/blog/ghostsocks-lummas-partner-in-proxy

Adversary Research - 23/02/25

Written by



Research Team - Infrawatch

5 min read

[Learn More](#)



This analysis explores GhostSocks, a Golang-based SOCKS5 backconnect proxy malware, detailing its integration with LummaC2 and its command-and-control infrastructure. We highlight its use of obfuscation and relay-based C2 communication, emphasising the threat it poses to financial institutions and other high-value targets.

[Register Infrawatch Beta Interest](#)

In this post:

Section

Introduction

GhostSocks, a Golang-based SOCKS5 backconnect proxy malware, was first identified in October 2023 when it was advertised on a Russian-language criminal forum, and supports Microsoft Windows alongside Linux. Its distribution expanded to English-speaking criminal forums in July 2024 in posts under the moniker "*GhostSocks*".

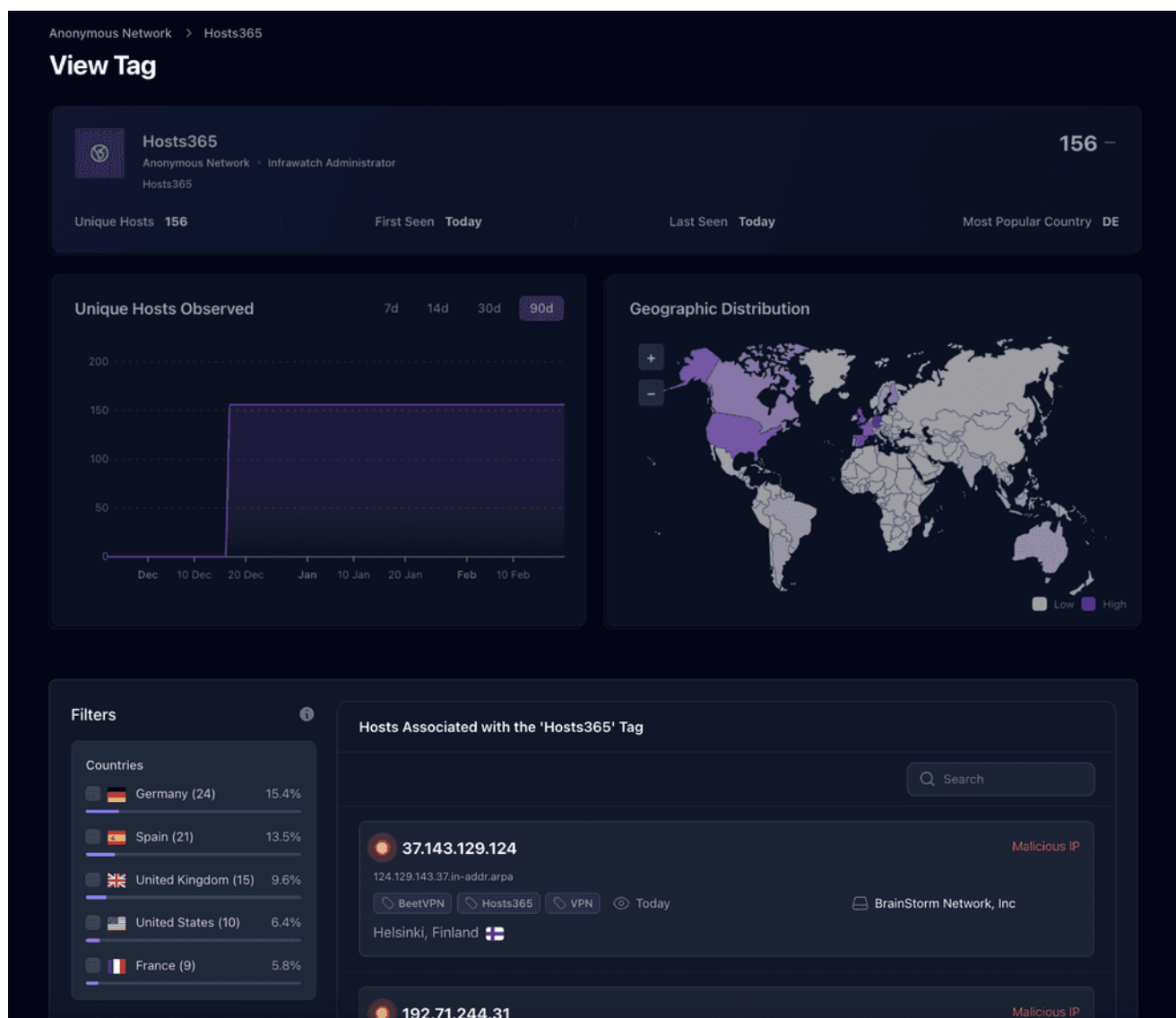
Primarily deployed alongside the *LummaC2* (Lumma) information stealer, *GhostSocks* is offered as a Malware-as-a-Service (MaaS), providing threat actors with an easily accessible tool to further monetise compromised systems.

The integration of *GhostSocks* with *Lumma*, facilitated by features like automatic provisioning and discounted pricing for Lumma users, highlights a deliberate effort to enhance post-infection capabilities, enabling heightened credential abuse and improving the likelihood of bypassing anti-fraud mechanisms.

This analysis delves into the workings of *GhostSocks*, exploring its close relationship with Lumma, its technical implementation, and its operational mechanisms.

Infrawatch's Residential Proxy Intelligence Data Feed

Infrawatch's Residential Proxy Feed offers real-time intelligence on commercially available residential proxies and VPNs, enabling organisations to proactively defend against fraud and abuse. This feed provides attributed, live data on infrastructure used by proxy services, allowing for the **identification**, **exploration**, and **blocking** of potentially risky hosts.



Lumma & GhostSocks - A Match Made In Moscow?

GhostSocks likely maintains a close direct relationship with Lumma's developer:

- On February 6, 2024, a partnership was announced via Telegram, introducing a new feature within the Lumma administration panel that enables the automatic provisioning of *GhostSocks* to Lumma infections.
- On July 22, 2024, the operator behind *GhostSocks* advertised the malware on a well-known English-speaking criminal forum, offering a substantial discount to customers already holding a Lumma license—further reinforcing its integration within the Lumma ecosystem.
- Furthermore, the malware employs two techniques that share similarities with Lumma: an anti-sandboxing methodology utilising the `GetCursorPos` API and a protection mechanism to prevent non-compiled builds from executing.

Any user can also sign up for GhostSocks independently, pay a fee of \$150 in Bitcoin, and build the malware themselves—this follows the Malware-as-a-Service (MaaS) model. The *GhostSocks* panel login can be observed below in Figure 1.

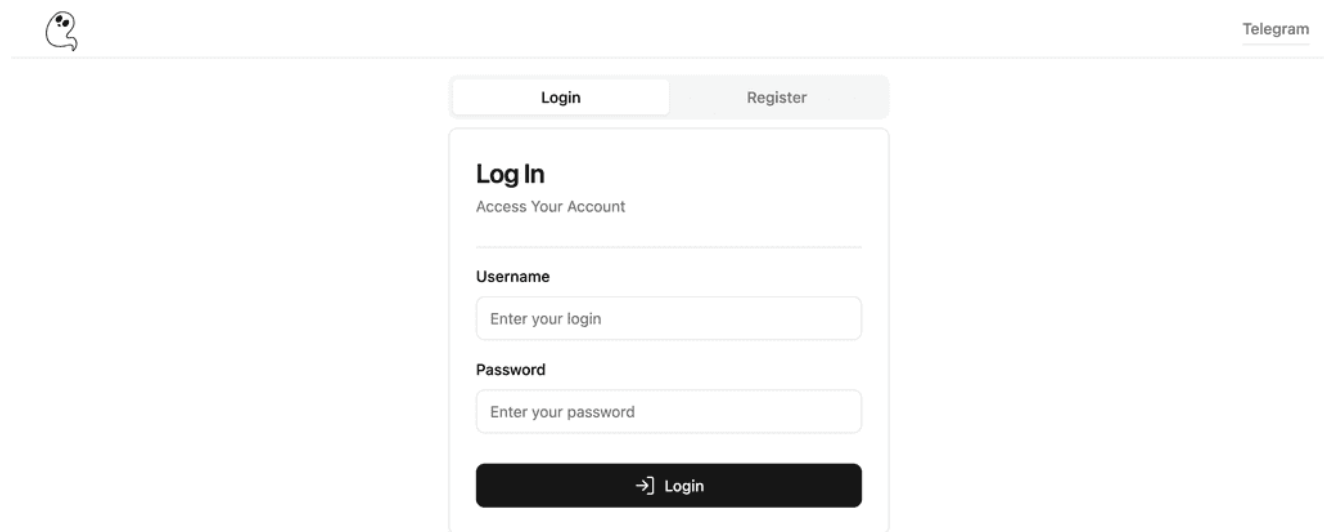


Figure 1 - *GhostSocks* MaaS Login Panel

Why?

The addition of a SOCKS5 backconnect feature to existing *Lumma* infections, or any malware for that matter, is highly lucrative for threat actors. By leveraging victims' internet connections, attackers can bypass geographic restrictions and IP-based integrity checks, particularly those enforced by financial institutions and other high-value targets. This capability significantly increases the probability of success for unauthorized access attempts using credentials harvested via infostealer logs, further enhancing the post-exploitation value of *Lumma* infections.

GhostSocks Malware

The basis of this analysis will focus on an *GhostSocks* sample observed on 15 February 2025 with the SHA-256 hash:
c92b21bdb91fe4c0590212e650212528a1f608a2ea086ce5eb5ac6d05edc41f7.

The above-mentioned *GhostSocks* sample is heavily obfuscated at points, likely making use of the popular open-source Go obfuscator [Garble](#). Along with some features from [Gofuscator](#) such as inline XOR-based string deobfuscation.

Upon initialisation, *GhostSocks* builds an embedded configuration structure comprised of hardcoded data and dynamically-calculated values. It is likely the hardcoded data changes on a per-build basis to distinct between different users of the MaaS:

```
{
  "buildVersion": "0pTk.PWh2DyJ", // <- likely an internal reference to the current
  build version
  "md5": "bb857552657a9c31e68797e9bd30ac2", // <- the MD5 hash of the malware on-
  disk, gathered from GetModuleHandle
  "proxyUsername": "uDoSfUGf", // <- the SOCKS5 back-connect username to be used
  "proxyPassword": "uDoSfUGf", // <- The SOCKS5 back-connect password to be used
  "userId": "gpn4wrgAehjlgkUKkN33e4iDkc10fRHA", // <- likely to identify the
  affiliate
}
```

Figure 2 - JSON configuration format

The configuration is then encoded into a JSON object, obfuscated, and written to %APPDATA%\config. A C2 IP and port is then deobfuscated and stored for later use in the C2 communication stage, in this instance: 46.8.232[.]106:3000.

Initial Beacon

GhostSocks uses a fairly simple relay-based C2 implementation using a simple HTTP API, in which an intermediary server sits in-between the real C2 and the victim. Most of the Tier 2 relays observed by Infrawatch communicate over port 3001.

Upon a victim first connecting to the C2, *GhostSocks* starts to build the HTTP GET query parameters derived from the configuration and a **X-Api-Key** header required for all requests to the C2.

Surprisingly, the "authentication" is simply a pseudo-random alphanumeric string with a length of 8 (e.g. Fm2qKy29: **^[A-Za-z0-9]{8}\$**) and does not rely on being derived from a value within the malware's configuration. The URI consists of values from the configuration, to the endpoint **/api/helper-first-register**. The full URI can be observed below in Figure 3.

```
/api/helper-first-register?
buildVersion=0pTk.PWh2DyJ&md5=bb857552657a9c31e68797e9bd30ac2&proxyPassword=uDoSfUGf&p
```

Figure 3 - Example initial beacon HTTP URI

If the **X-Api-Key** is not present, the C2 responds with the HTTP body: **Forbidden: Invalid API Key**. A normal beacon response from the C2 can be observed below in Figure 4:

```
185.21.13[.]144;30820;[obfuscated_blob_data]
```

Figure 4 - GhostSocks initial beacon response

As can be observed above, an IP and port pair exists - a Tier 1 node in which the SOCKS5 back-connect takes place. Over the period of this analysis, Infrawatch managed to identify three different unique back-connect hosts being used by *GhostSocks*:

```
195.200.28[.]33:27799
212.34.130[.]72:15701
185.21.13[.]144:15706
185.21.13[.]144:30820
```

Figure 5 - SOCKS5 Backconnect hosts

The port used in the backconnect hosts is possibly assigned based on the affiliate, as these are shared among all *GhostSocks* customers. However, this cannot be confirmed at the time of writing.

A TCP connection is then established with the returned IP and port pair, and a SOCKS5 backconnect tunnel is set up using the credentials from the configuration.

Additional C2s were discovered by Infrawatch over a 2-month period, which can be observed in Figure 6.

```
91.142.74[.]28
195.200.28[.]33
195.200.31[.]22
185.245.106[.]67
77.238.224[.]56
77.238.245[.]11
185.121.233[.]152
46.8.232[.]106
38.180.61[.]247
46.8.236[.]61
185.157.213[.]253
77.238.245[.]233
195.2.70[.]38
91.212.166[.]91
```

Figure 6 - Additional GhostSocks C2s & Backconnect Hosts

The majority of the C2s used by *GhostSocks* and backconnect sit on VDSina (AS216071) - note that Russian-speaking server providers use Virtual Dedicated Server (VDS) in place of Virtual Private Server (VPS). VDSina is officially registered as Servers Tech Fzco, which is a company officially registered in the United Arab Emirates. AS216071 is also home to several commercial VPNs such as VydraVPN, VPNSurf, PabloVPN and more.

Additional Backdoor Functionality

GhostSocks also contains additional backdoor functionality, beyond the primary SOCKS5 backconnect proxy capability. Some of the additional functionality, along with their internal name and command ID, includes:

1. Arbitrary Command Execution (*shell*, ID: 5):
 1. Executes arbitrary commands sent by the C2: `cmd.exe /C <command>`
2. Modification Of SOCKS5 Credentials - (ID: 4)
 1. Ability to add or remove new SOCKS5 backconnect credentials for the bot, the credentials are parsed within a string with the delimiter ":"
3. Download & Execute Arbitrary Executables (*update*, ID: 6)
 1. Download an arbitrary executable, execute it using the same code used for the *shell* command, with the path as the parameter

What is SOCKS5 Backconnect?

SOCKS5 is a proxy protocol that facilitates the routing of network traffic through an intermediary server. In a **backconnect** setup, instead of a client connecting directly to a proxy server, the proxy server (in this case, the infected machine) **initiates** the connection to the attacker-controlled infrastructure. This allows threat actors to use the compromised system as a relay, effectively masking their true origin while interacting with external services.

How GhostSocks Uses it

1. **C2 Response** – The malware queries its C2, which returns a **Tier 1 relay IP and port**.
2. **Connection Establishment** – *GhostSocks* **initiates a TCP connection** to this Tier 1 node.
3. **SOCKS5 Tunnel Creation** – The malware sets up a **SOCKS5 proxy tunnel**, allowing attackers to route their traffic through the infected system.
4. **Credential Abuse & Evasion** – By leveraging this connection, threat actors can interact with online services using the victim's IP address, bypassing **geolocation restrictions, fraud detection mechanisms, and IP-based security controls** (commonly used by financial institutions).

Signaturing GhostSocks' C2s

Infrawatch provides IP-level attribution for residential proxies and VPNs, covering over 400 commercial services. Additionally, we track over 130 distinct malware families. Integrating *GhostSocks* C2 tracking was a straightforward task for our Research Team, enabling customers to proactively block C2 infrastructure in anticipation of potential malicious activity on their network.

As mentioned before, most of the Tier 2 servers use the port **30001**, and requests emitting the **X-Api-Key** HTTP header result in an error message. Upon inspection of other C2s, the headers are persistent for responses, which render a hash of:

86362ac6d972b1b55f1f434811d014316196f0e193878d8270dae939efb25908

Using Infrawatch's YARA signature detection capabilities, we can craft a rule to track the C2s in our internet-wide scans of this port:

```
import "infrawatch"

rule GhostSocks
{
  condition:
    infrawatch.http.port == 30001 and
    infrawatch.http.body == "Forbidden: Invalid API Key" and
    infrawatch.http.status_code == 403 and
    infrawatch.http.headers_hash ==
"86362ac6d972b1b55f1f434811d014316196f0e193878d8270dae939efb25908"
```

Conclusion

GhostSocks exemplifies the continued commodification of SOCKS5 backconnect malware within the criminal ecosystem. While backconnect proxies are not a new technique, *GhostSocks'* seamless integration with **LummaStealer** and its availability through a **Malware-as-a-Service (MaaS) model** make it the obvious choice for a threat actor to use in a bid to monetise their victims to the maximum.

By leveraging C2 behavioural indicators—such as consistent X-Api-Key error responses—defenders can more effectively track and prevent *GhostSocks* C2 infrastructure from posing a threat within their estate.

In addition to tracking malicious infrastructure, we also provide real-time, attributed intelligence on **legitimate residential proxy and VPN providers**.

IOCs

91.142.74[.]28
195.200.28[.]33
195.200.31[.]22
185.245.106[.]67
77.238.224[.]56
77.238.245[.]11
185.121.233[.]152
46.8.232[.]106
38.180.61[.]247
46.8.236[.]61
185.157.213[.]253
77.238.245[.]233
195.2.70[.]38
91.212.166[.]91
195.200.28[.]33
212.34.130[.]72
185.21.13[.]144

