INTRINSEC

Cyber Threat Intelligence

Various actors actively deploying Lumma Stealer in multiple campaigns

Follow us for more Cyber Threat Intelligence content

www.intrinsec.com	www.intrinsec.com/blog	@Intrinsec	@Intrinsec

Table of contents

Key findings

Lumma Stealer, also known as LummaC2 Stealer, is a malware-as-a-service sold through Telegram and Russian-speaking cybercrime forums. In this report, the following will be addressed:

- The presence of Lumma in Russian-speaking forums and Telegram.
- Code analysis of different campaigns distributing Lumma stealer using various techniques.
- The infrastructure associated with Lumma stealer, including the old and new versions of C2 panels.
- A trail, that we uncovered, which indicates a potential use of Lumma by a Russian intrusion set.

Introduction

Lumma is an information stealer written in C/C++ language that has been observed in the wild since at least August 2022, while the first Lumma sample was seen on Malware Bazaar on December 20, 2022. The user “**Lumma**” started to advertise the stealer on Russian-speaking forums by the end of 2022 and a Telegram channel with around 1 700 subscribers as of September 2023 was also created to advertise it. Additionally, a website currently offline where users could buy the stealer for a price ranging between $250 to $1000 was also promoted.

The stealer is believed to have been developed by the threat actor "**Shamel**", who goes by the alias "**Lumma**". This malware is designed to pilfer sensitive data from infected devices. Among the data targeted are cryptocurrency wallets, browser extensions, two-factor authentication codes, logins and passwords stored on browsers, and various files. Once the targeted data is obtained, it is exfiltrated to a C2 server and either sold privately or on various marketplaces such as Russian Market. The developer of the malware has consistently introduced improved iterations of its payload since its first public release, including a loader module. The malware was usually deployed through fake installers of cracked software. On September 27, the researcher ExecuteMalware shared on Github a list of IoCs related to a SocGholish infection delivering the Lumma stealer. This incident makes Lumma a very credible threat, as SocGholish is the loader associated with EvilCorp.

I - Strategical Intelligence

1. Victimology

Lumma logs were first observed in Russian Market in April 2023, gaining a higher share of the total logs in this marketplace since then, with 50% of all logs put for sale in the last 3 months, and 20% year to date. Around 410 000 Lumma logs were put for sale on Russian Market.

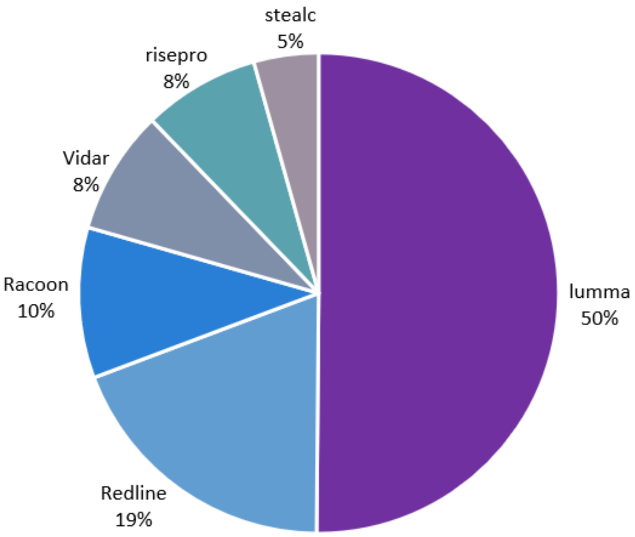
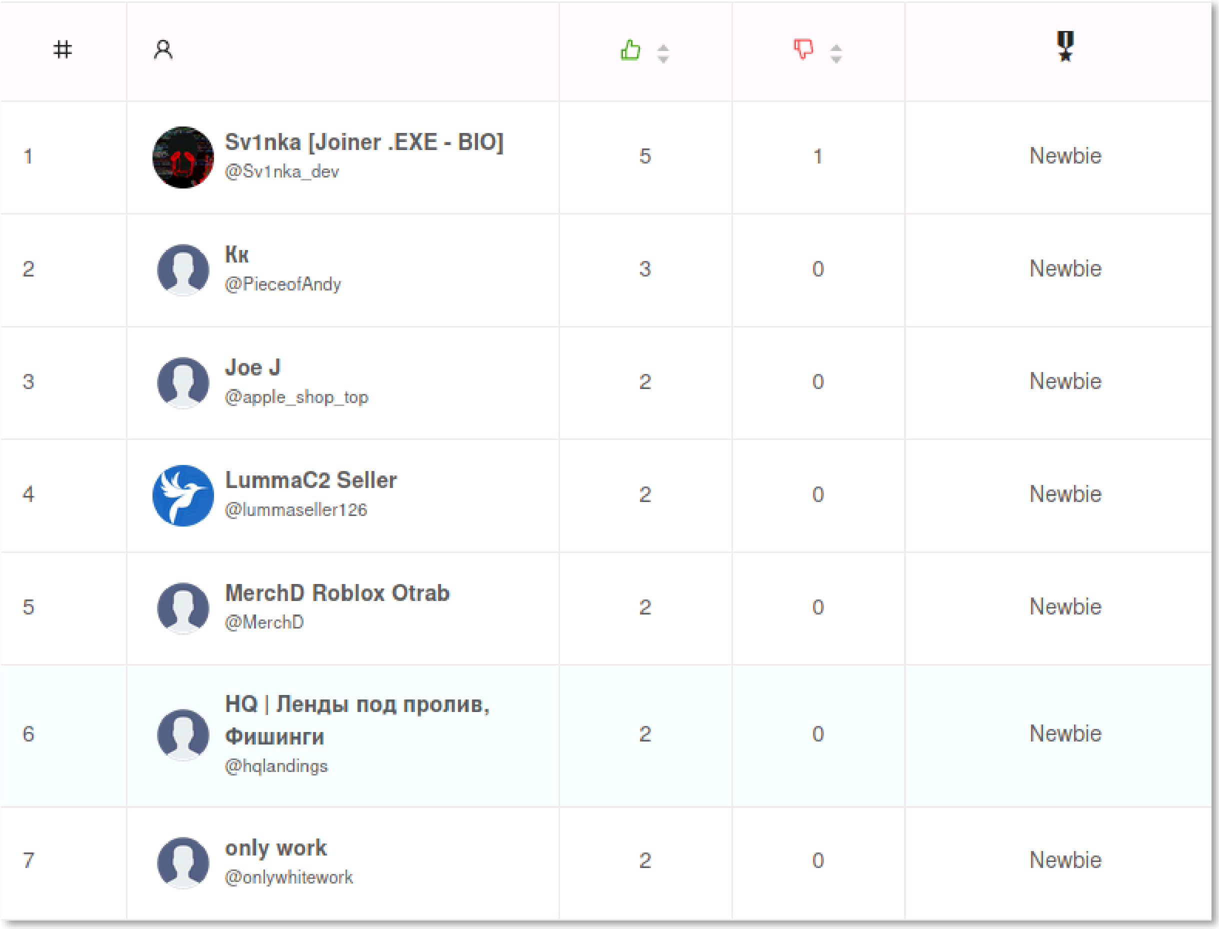


Figure 1 : Graph displaying the percentage of logs sold on Russian Market per Stealer for the last 3 months.

2. Attribution

The user “**Lumma**” (also known as “**Shamel**”) employs primarily Russian as the speaking language and regularly stated that he does not want to speak in English. Various Telegram channels are associated to Lumma. The main one, “**@LummaC2Stealer**” is the one with most subscribers and is used to advertise the stealers’ updates and various information, including advertisement for other tools. The admin of this channel is “**@lummaseller126**”, who is mentionned as the account to contact to buy the stealer. There is also a chat group named “**LummaC2Team**”, with around 800 members, mainly for users to ask questions. Finally, the channel “**LummaC2Blacklist**”, with 31 subscribers, is used to expose scammers trying to pass as resellers of Lumma.

Using the Telegram channel analysis tool Chatkeeper, we can get an overview of the reputations of users of the group chat “**LummaC2Team**”. The user “**sv1nka_dev**” is the most upvoted user. Viewing his messages on the channel, he acts like a secondary admin, redirecting users to the seller and the “real” admin and shows a real knowledge of how the channel is set up. He is also a member of the channels “**Ransom Team | chat**”, “**Crypt Service | Услуги Крипта**” and “**Stealer Developers**”. While this user may not be associated to Lumma’s development, since he stated in a message that he was not a “lumma coder”, his knowledge and popularity among this channel makes him a threat actor of interest who may develop tools or collaborate in already existing or future cybercrime operations.



#	👤	👍	👎	🎖
1	Sv1nka [Joiner .EXE - BIO] @Sv1nka_dev	5	1	Newbie
2	Кк @PieceofAndy	3	0	Newbie
3	Joe J @apple_shop_top	2	0	Newbie
4	LummaC2 Seller @lummaseller126	2	0	Newbie
5	MerchD Roblox Otrab @MerchD	2	0	Newbie
6	HQ | Ленды под пролив, Фишинги @hqlandings	2	0	Newbie
7	only work @onlywhitework	2	0	Newbie

Figure 2 : Users of the Telegram chat "LummaC2Team" with the most favorable reputation, using Chatkeeper's Telegram channel analysis tool. https://client.chatkeeper.app/chat/-1001944757940/reputation/.

On the dedicated thread for Lumma on XSS, where the seller/coder announces the updates, there is a user with the alias "**dark_dream**" who presents himself as a member of the developers of LC2 (Lumma C2) and who also participated in developing the "**Morpher**". The Morpher would be the tool/code responsible for obfuscation. According to the user "**Lumma**", the Morpher has the following capabilities:

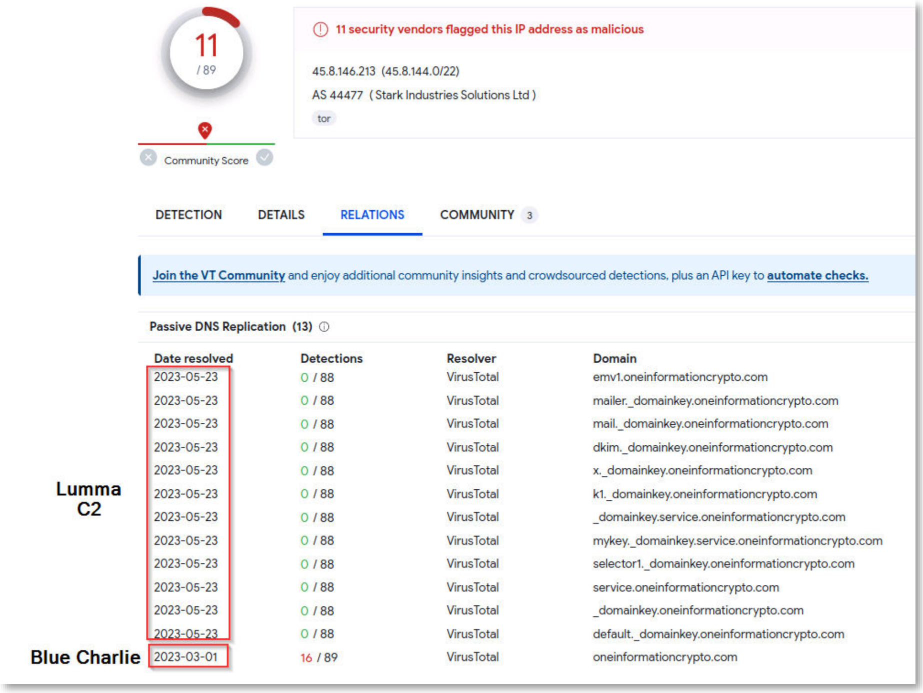
- Create clones of original code blocks, that are mixed with garbage code. Branches of conditional transitions lead to them, leading to dead code.
- Original code is mixed with garbage that does not disturb the main code. If the garbage is removed, the code will not work.
- Constants are replaced by various arithmetic expressions that compute these constants in a random, multi-step fashion.
- The transitions and branching are broken down into additional conditional transitions that include jumps to random garbage code blocks that can be executed or not.
- Garbage block can have jumps to other garbage block as well as the original code.
- Several passes are added for obfuscation and new passes break the signature.
- The Morpher is fully automated and allows to remorph at will, creating unique builds each time.

According to "**dark_dream**", the team spent 2-3 years to develop this tool. A reverse engineer under the alias "**waahoo**" published an analysis of the Morpher on XSS. While the result does not give us any new insights about the Morpher, "**dark dream**" is seen praising the work of the reverser. Indeed, he mentions that the Morpher has "**two main objectives**": complicate reverse work and evade antiviruses. Moreover, "**dark_dream**" says that he will soon "write an article" about his Morpher as an introduction for beginners.

2.1 Is Lumma C2 only used by cybercriminals?

In this analysis, we found that several Lumma C2 panel's IP addresses belong to "**Stark Industries Solutions Ltd**". Two of these IP, 45.8.146[.]213 and 45.8.146[.]227 (and their respective domains **stateinfospace**[.]**com** and **oneinformationcrypto**[.]**com**) are mentioned in a report by RecordedFuture as infrastructure related to the Russia-nexus group BlueCharlie (aka Callisto, SEABORGIUM). This intrusion set was previously observed in various credentials harvesting campaigns, using phishing websites with fake forms and login pages. While this information does not show a direct link between Lumma and BlueCharlie, as Stark Industries may have reattributed some IP addresses, the interest in credentials shown by this intrusion set might explain the choice of using a stealer. RecordedFuture's report indicates that the intrusion set registered a wave of domains, including those mentioned above, around March 2023, which is consistent with when the domains associated to these IPs were registered. According to Fofa, these domains/IPs were associated with a Lumma C2 panel since at least end of May 2023, as the file "**Doberman.min.js"** was found inside it at that time, so only two/three months after them being potentially registered by BlueCharlie. This is also confirmed by some urlscans dating from May 7 2023, exposing the Lumma C2 panel login page. As the RecordedFuture report was published in August 2023, we cannot suspect that they had to change the ownership of these IPs so quickly.

As such we can state, with a low confidence, the hypothesis that BlueCharlie uses, used, or planned to use Lumma Stealer for some of its operations, which is further supported by the fact that these IPs have not registered a domain rotation since March 2023. It is important to note that this link can also be based on an error from RecordedFuture, as these domains' attribution to BlueCharlie is solely based on the naming pattern, the use of specific registrar, and a precise convention for the domain's certificate.



11 security vendors flagged this IP address as malicious

45.8.146.213 (45.8.144.0/22)
AS 44477 (Stark Industries Solutions Ltd)
tor

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Passive DNS Replication (13)

	Date resolved	Detections	Resolver	Domain
Lumma C2	2023-05-23	0 / 88	VirusTotal	emv1.oneinformationcrypto.com
	2023-05-23	0 / 88	VirusTotal	mailer._domainkey.oneinformationcrypto.com
	2023-05-23	0 / 88	VirusTotal	mail._domainkey.oneinformationcrypto.com
	2023-05-23	0 / 88	VirusTotal	dkim._domainkey.oneinformationcrypto.com
	2023-05-23	0 / 88	VirusTotal	x._domainkey.oneinformationcrypto.com
	2023-05-23	0 / 88	VirusTotal	k1._domainkey.oneinformationcrypto.com
	2023-05-23	0 / 88	VirusTotal	_domainkey.service.oneinformationcrypto.com
	2023-05-23	0 / 88	VirusTotal	mykey._domainkey.service.oneinformationcrypto.com
	2023-05-23	0 / 88	VirusTotal	selector1._domainkey.oneinformationcrypto.com
	2023-05-23	0 / 88	VirusTotal	service.oneinformationcrypto.com
	2023-05-23	0 / 88	VirusTotal	_domainkey.oneinformationcrypto.com
	2023-05-23	0 / 88	VirusTotal	default._domainkey.oneinformationcrypto.com
Blue Charlie	2023-03-01	16 / 89	VirusTotal	oneinformationcrypto.com

Figure 3 : https://www.virustotal.com/gui/ip-address/45.8.146.213/relations.

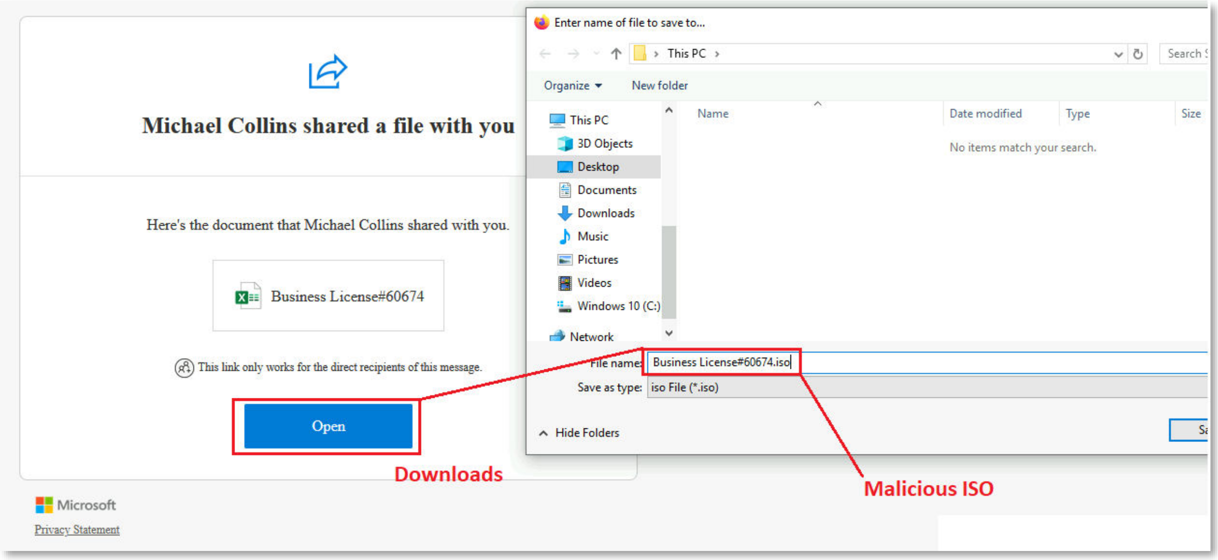
Recently, we observed a trend with Russian intrusion sets potentially relying more on cybercrime tools for their operations. In our report from July 2023 (*"UAC-0006, a hybrid SmokeLoader campaign?"*), we discussed a large SmokeLoader phishing campaign against Ukrainian entities that may be linked to the Russian intrusion set UAC-0006, and in July 2022, RecordedFuture revealed that UAC-0113, an intrusion set which could be linked to Sandworm, has been using HTML smuggling to deploy two malwares sold and used by cybercriminals, Colibri Loader and Warzone RAT. If this trend continues, as it is advantageous and cheaper for intrusion sets to use already developed sophisticated malware, it might prove to be more difficult to separate activities related to the Russian cybercrime ecosystem and Russian intrusion sets. Analysing the goals and victimology of campaigns leveraging cybercrime tools might be useful to discriminate between state-sponsored activities and financially oriented ones.

II - Tactical Intelligence

1. Tactics, Techniques and Procedures

1.1 HTML smuggling campaign operated by botnet ID “PrTi07—test2”

This campaign distributed HTML files dubbed “**Business Licence#60674.html**”, “**Balance Sheet#37553.html**”, “**Employee Contract_14212.html**” and “**Request for Proposal (RFP)#51982.html**”. All four delivering ISO disks with the same names.



Michael Collins shared a file with you

Here's the document that Michael Collins shared with you.

Business License#60674

This link only works for the direct recipients of this message.

Enter name of file to save to...

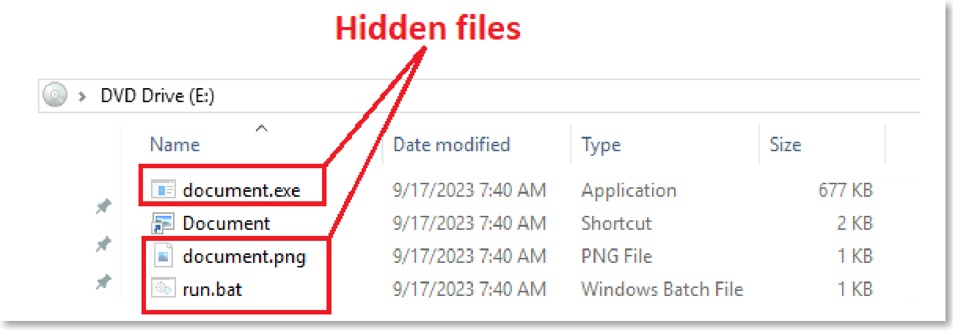
File name: Business License#60674.iso

Save as type: iso File (*.iso)

No items match your search.

Figure 4 : Content of the HTML file.

Inside the ISO can be found 3 files in “hidden mode” : an EXE application, a PNG, and a Batch script. The only visible file is a shortcut with an icon of a picture.

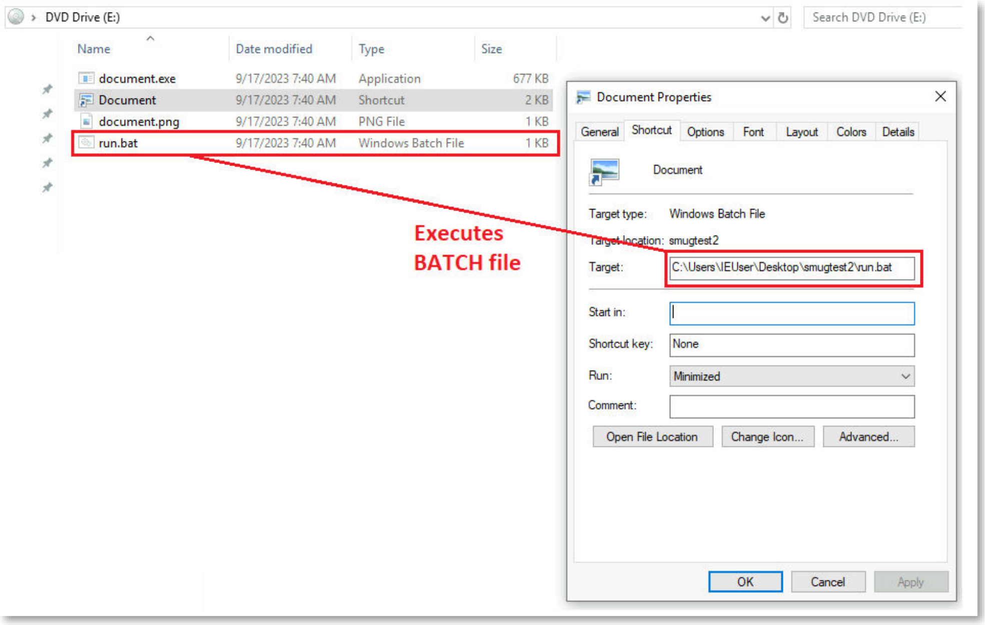


DVD Drive (E:)

Name	Date modified	Type	Size
document.exe	9/17/2023 7:40 AM	Application	677 KB
Document	9/17/2023 7:40 AM	Shortcut	2 KB
document.png	9/17/2023 7:40 AM	PNG File	1 KB
run.bat	9/17/2023 7:40 AM	Windows Batch File	1 KB

Figure 5 : Content of the ISO file.

The goal of the shortcut is to execute the hidden Batch script.



DVD Drive (E:)

Name	Date modified	Type	Size
document.exe	9/17/2023 7:40 AM	Application	677 KB
Document	9/17/2023 7:40 AM	Shortcut	2 KB
document.png	9/17/2023 7:40 AM	PNG File	1 KB
run.bat	9/17/2023 7:40 AM	Windows Batch File	1 KB

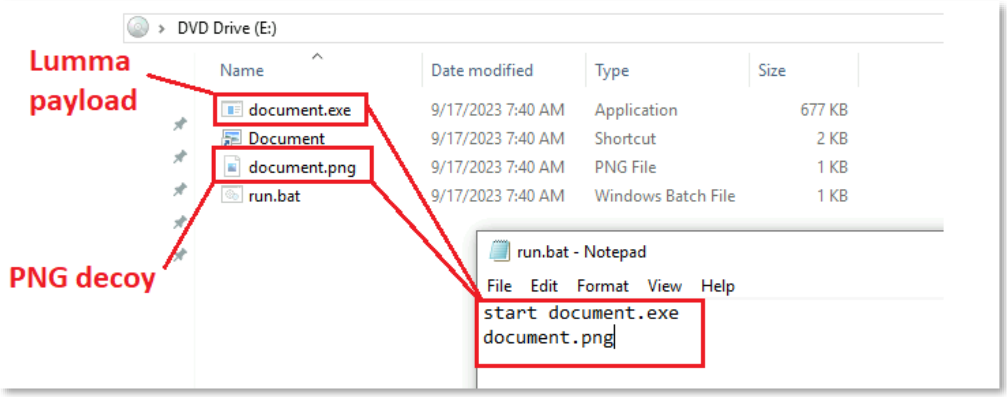
Document Properties

Document

Target type: Windows Batch File
Target location: smugtest2
Target: C:\Users\IEUser\Desktop\smugtest2\run.bat
Start in:
Shortcut key: None
Run: Minimized
Comment:

Figure 6 : Command line in the shortcut that launches the Batch script.

The Batch script then launches both the EXE application and the PNG.



DVD Drive (E:)

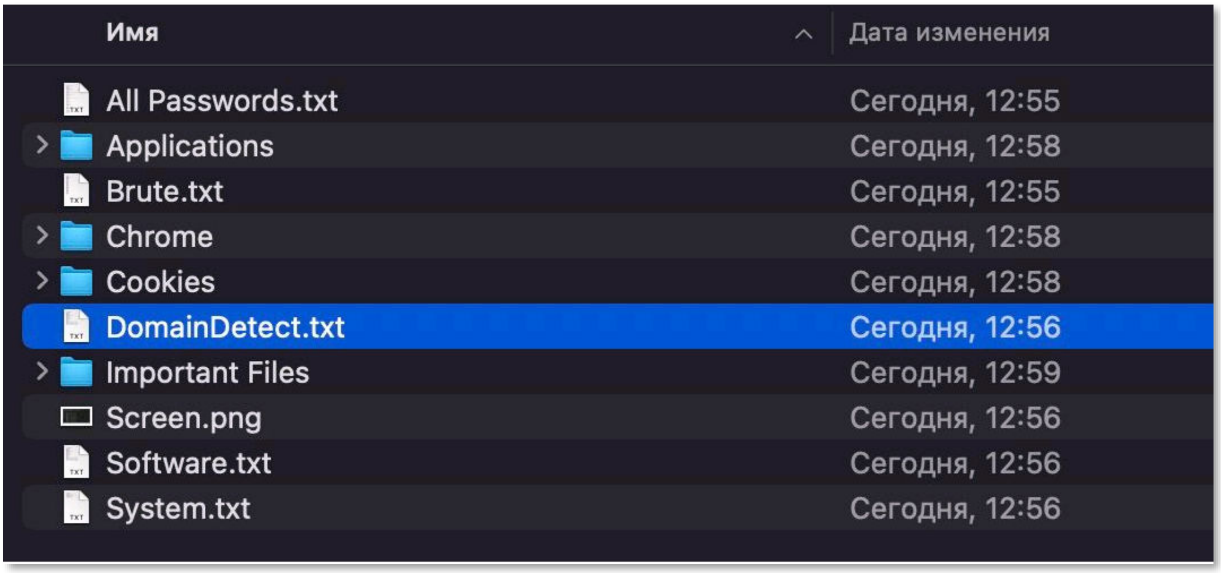
Name	Date modified	Type	Size
document.exe	9/17/2023 7:40 AM	Application	677 KB
Document	9/17/2023 7:40 AM	Shortcut	2 KB
document.png	9/17/2023 7:40 AM	PNG File	1 KB
run.bat	9/17/2023 7:40 AM	Windows Batch File	1 KB

run.bat - Notepad

```
start document.exe
document.png
```

Figure 7 : Command in the Batch script that launches the last two files.

Once launched, the EXE will collect files and login information from the infected machine and exfiltrate them to one of the two C2 servers that are stored in its configuration.

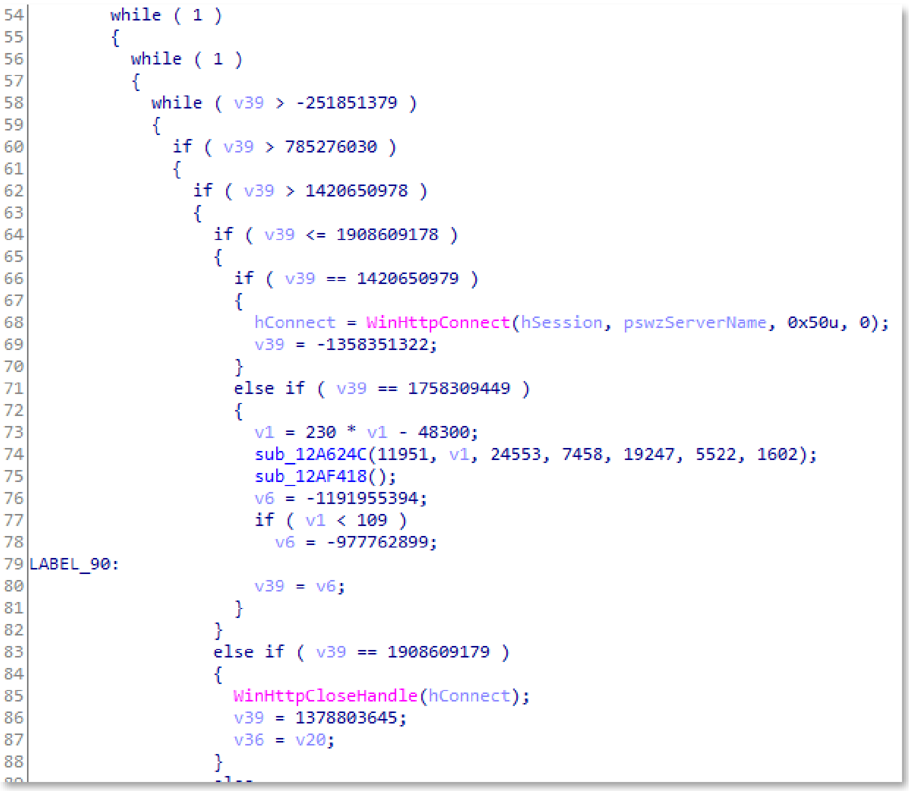


Имя	Дата изменения
All Passwords.txt	Сегодня, 12:55
Applications	Сегодня, 12:58
Brute.txt	Сегодня, 12:55
Chrome	Сегодня, 12:58
Cookies	Сегодня, 12:58
DomainDetect.txt	Сегодня, 12:56
Important Files	Сегодня, 12:59
Screen.png	Сегодня, 12:56
Software.txt	Сегодня, 12:56
System.txt	Сегодня, 12:56

Figure 8 : Content of a Lumma log as shared by the user "Lumma" on XSS.

2. Code Analysis

As described by eSentire in their blog post, the code is highly obfuscated using Control Flow Flattening (CFF), which makes reversing highly time consuming. This is probably what the developer talked about while mentioning the "**morpher**". In addition, it contains over 1200 functions.



```
        while ( 1 )
        {
          while ( 1 )
          {
            while ( v39 > -251851379 )
            {
              if ( v39 > 785276030 )
              {
                if ( v39 > 1420650978 )
                {
                  if ( v39 <= 1908609178 )
                  {
                    if ( v39 == 1420650979 )
                    {
                      hConnect = WinHttpConnect(hSession, pswzServerName, 0x50u, 0);
                      v39 = -1358351322;
                    }
                    else if ( v39 == 1758309449 )
                    {
                      v1 = 230 * v1 - 48300;
                      sub_12A624C(11951, v1, 24553, 7458, 19247, 5522, 1602);
                      sub_12AF418();
                      v6 = -1191955394;
                      if ( v1 < 109 )
                        v6 = -977762899;
LABEL_90:
                      v39 = v6;
                    }
                  }
                  else if ( v39 == 1908609179 )
                  {
                    WinHttpCloseHandle(hConnect);
                    v39 = 1378803645;
                    v36 = v20;
                  }
```

Figure 9: Heavily obfuscated function with CFF.

The function above shows how the obfuscation works with lots of non-useful loops and conditions. The graph view of the full function can help illustrate how the morpher obfuscates. The added control flow hides the actual meaningful instructions.

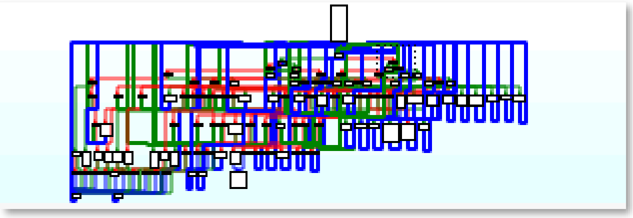
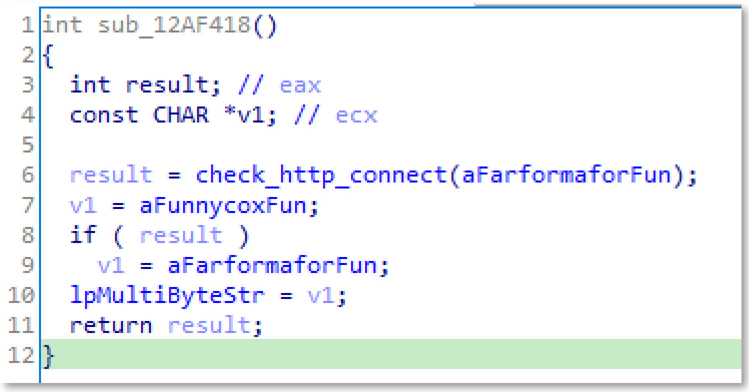


Figure 10: Graph view of an CFF obfuscated function.

This function is actually a network test, with the malware trying to connect to the first C&C server from the list in the configuration located in the *.rdata* section.



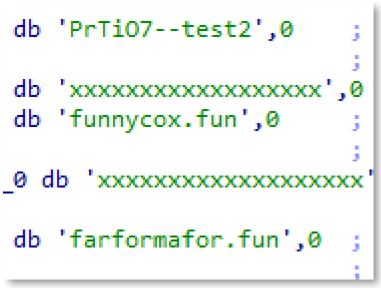
```
int sub_12AF418()
{
  int result; // eax
  const CHAR *v1; // ecx

  result = check_http_connect(aFarformaforFun);
  v1 = aFunnycoxFun;
  if ( result )
    v1 = aFarformaforFun;
  lpMultiByteStr = v1;
  return result;
}
```

Figure 11: Checks if it can connect to the first C&C domain.

The function above shows the connection attempt to the first C&C domain, returns it if successful and moves to the second one if it failed.

We were able to locate the configuration containing bot id and C&C from the *.rdata* section:

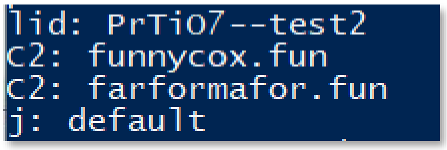


```
 db 'PrTiO7--test2',0    ;
                         ;
 db 'xxxxxxxxxxxxxxxxxxx',0
 db 'funnycox.fun',0     ;
                         ;
_0 db 'xxxxxxxxxxxxxxxxxxxx'

 db 'farformafor.fun',0  ;
                         ;
```

Figure 12: Configuration located in the .rdata section.

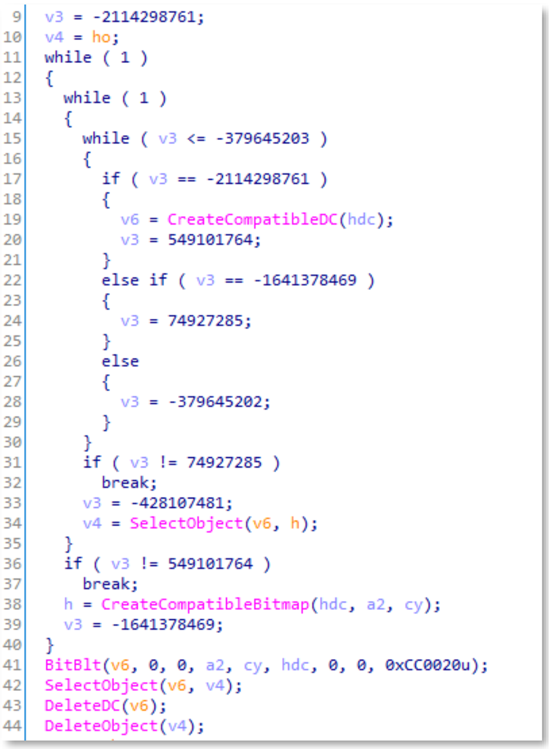
The configuration can also be extracted using this tool developed by the threat analyst RussianPanda.



```
lid: PrTiO7--test2
C2: funnycox.fun
C2: farformafor.fun
j: default
```

Figure 13: Configuration extracted using RussianPanda's tool.

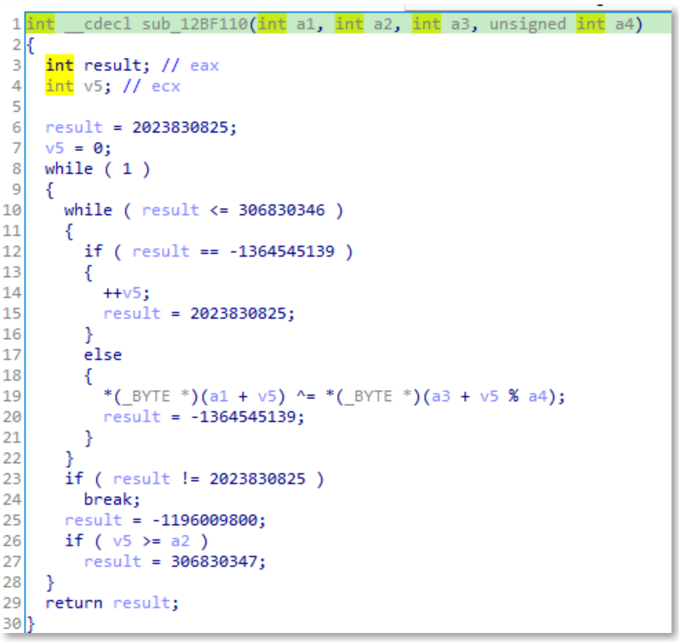
We located what may be a screenshotting functionality using GetDC, BitBlt and CreateCompatibleDC windows APIs.



```
v3 = -2114298761;
v4 = ho;
while ( 1 )
{
  while ( 1 )
  {
    while ( v3 <= -379645203 )
    {
      if ( v3 == -2114298761 )
      {
        v6 = CreateCompatibleDC(hdc);
        v3 = 549101764;
      }
      else if ( v3 == -1641378469 )
      {
        v3 = 74927285;
      }
      else
      {
        v3 = -379645202;
      }
    }
    if ( v3 != 74927285 )
      break;
    v3 = -428107481;
    v4 = SelectObject(v6, h);
  }
  if ( v3 != 549101764 )
    break;
  h = CreateCompatibleBitmap(hdc, a2, cy);
  v3 = -1641378469;
}
BitBlt(v6, 0, 0, a2, cy, hdc, 0, 0, 0xCC0020u);
SelectObject(v6, v4);
DeleteDC(v6);
DeleteObject(v4);
```

Figure 14: Function containing screenshotting capabilities.

In addition to CFF, Lumma uses multiple and different ciphers such as this simple XOR cipher to obfuscate strings.

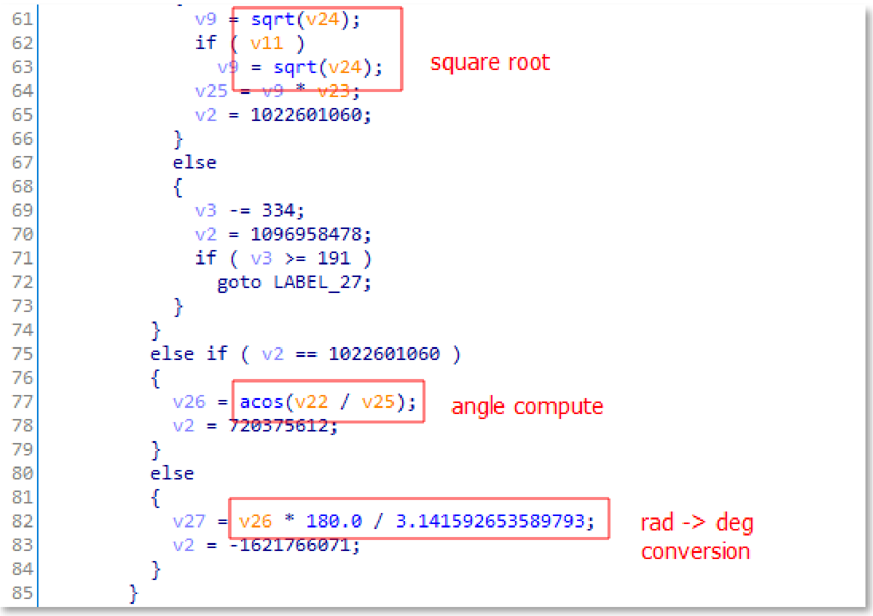


```
int __cdecl sub_12BF110(int a1, int a2, int a3, unsigned int a4)
{
  int result; // eax
  int v5; // ecx

  result = 2023830825;
  v5 = 0;
  while ( 1 )
  {
    while ( result <= 306830346 )
    {
      if ( result == -1364545139 )
      {
        ++v5;
        result = 2023830825;
      }
      else
      {
        *(_BYTE *)(a1 + v5) ^= *(_BYTE *)(a3 + v5 % a4);
        result = -1364545139;
      }
    }
    if ( result != 2023830825 )
      break;
    result = -1196009800;
    if ( v5 >= a2 )
      result = 306830347;
  }
  return result;
}
```

Figure 15: CFF obfuscated XOR cipher.

We found some interesting calculations such as square root, trigonometry, and angle conversion. We were not able to determine the purpose of these calculations, but we can assume that they may be used to add obfuscation layers.



```
          v9 = sqrt(v24);
          if ( v11 )
            v9 = sqrt(v24);
          v25 = v9 * v23;
          v2 = 1022601060;
        }
        else
        {
          v3 -= 334;
          v2 = 1096958478;
          if ( v3 >= 191 )
            goto LABEL_27;
        }
      }
      else if ( v2 == 1022601060 )
      {
        v26 = acos(v22 / v25);
        v2 = 720375612;
      }
      else
      {
        v27 = v26 * 180.0 / 3.141592653589793;
        v2 = -1621766071;
      }
    }
```

Figure 16: Mathematical expressions contained in a function with an unknown purpose.

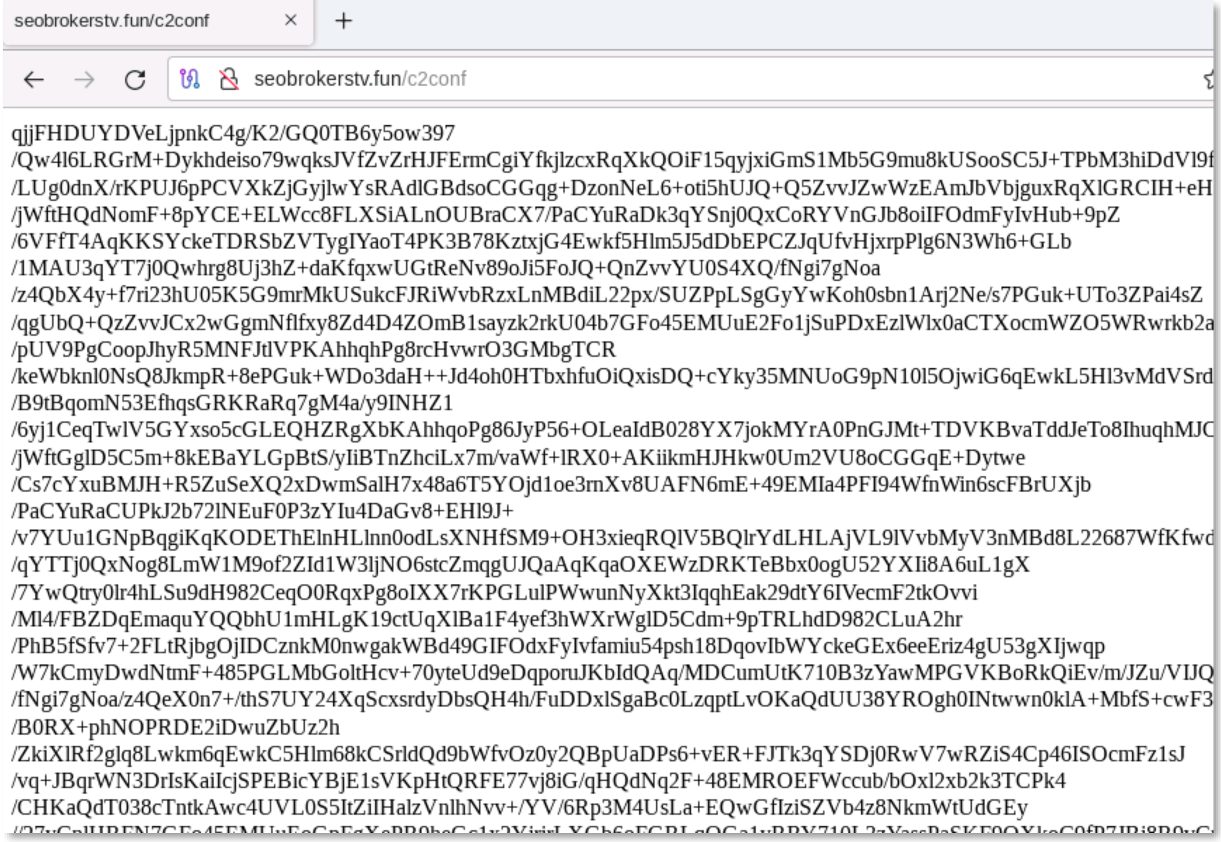
3. Infrastructure Analysis

3.1 Old version of Lumma stealer

The communications with the C2 server rely on GET and POST requests using the User-Agent "`TeslaBrowser/5.5`". The C2 endpoints are:

- `/c2conf` for retrieving the C2 configuration (GET request)
- `/c2sock` for data exfiltration (POST request)
- `/login` for access to the C2 panel

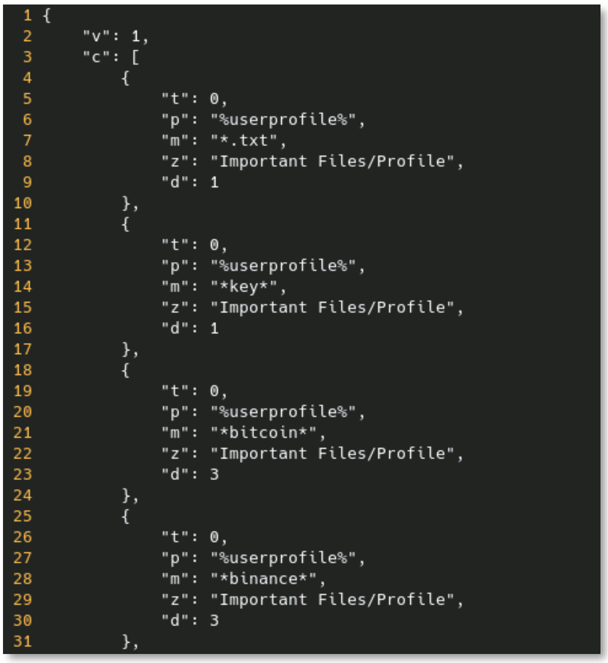
As some threat actors still run older versions of the stealer/panel, we were able to find a c2conf inside a Lumma panel:



```
qjjFHDUYDVeLjpnkC4g/K2/GQ0TB6y5ow397
/Qw4l6LRGrM+Dykhdeiso79wqksJVfZvZrHJFErmCgiYfkjlzcxRqXkQOiF15qyjxiGmS1Mb5G9mu8kUSooSC5J+TPbM3hiDdVl9f
/LUg0dnX/rKPUJ6pPCVXkZjGyjlwYsRAdlGBdsoCGGqg+DzonNeL6+oti5hUJQ+Q5ZvvJZwWzEAmJbVbjguxRqXlGRCIH+eH
/jWftHQdNomF+8pYCE+ELWcc8FLXSiALnOUBraCX7/PaCYuRaDk3qYSnj0QxCoRYVnGJb8oiIFOdmFyIvHub+9pZ
/6VFfT4AqKKSYckeTDRSbZVTygIYaoT4PK3B78KztxjG4Ewkf5Hlm5J5dDbEPCZJqUfvHjxrpPlg6N3Wh6+GLb
/1MAU3qYT7j0Qwhrg8Uj3hZ+daKfqxwUGtReNv89oJi5FoJQ+QnZvvYU0S4XQ/fNgi7gNoa
/z4QbX4y+f7ri23hU05K5G9mrMkUSukcFJRiWvbRzxLnMBdiL22px/SUZPpLSgGyYwKoh0sbn1Arj2Ne/s7PGuk+UTo3ZPai4sZ
/qgUbQ+QzZvvJCx2wGgmNflfxy8Zd4D4ZOmB1sayzk2rkU04b7GFo45EMUuE2Fo1jSuPDxEzlWlx0aCTXocmWZO5WRwrkb2a
/pUV9PgCoopJhyR5MNFJtlVPKAhhqhPg8rcHvwrO3GMbgTCR
/keWbknl0NsQ8JkmpR+8ePGuk+WDo3daH++Jd4oh0HTbxhfuOiQxisDQ+cYky35MNUoG9pN10l5OjwiG6qEwkL5Hl3vMdVSrd
/B9tBqomN53EfhqsGRKRaRq7gM4a/y9INHZ1
/6yj1CeqTwlV5GYxso5cGLEQHZRgXbKAhhqoPg86JyP56+OLeaIdB028YX7jokMYrA0PnGJMt+TDVKBvaTddJeTo8IhuqhMJC
/jWftGglD5C5m+8kEBaYLGpBtS/yIiBTnZhciLx7m/vaWf+lRX0+AKiikmHJHkw0Um2VU8oCGGqE+Dytwe
/Cs7cYxuBMJH+R5ZuSeXQ2xDwmSalH7x48a6T5YOjd1oe3rnXv8UAFN6mE+49EMIa4PFI94WfnWin6scFBrUXjb
/PaCYuRaCUPkJ2b72lNEuF0P3zYIu4DaGv8+EHl9J+
/v7YUu1GNpBqgiKqKODEThElnHLlnn0odLsXNHfSM9+OH3xieqRQlV5BQlrYdLHLAjVL9lVvbMyV3nMBd8L22687WfKfwd
/qYTTj0QxNog8LmW1M9of2ZId1W3ljNO6stcZmqgUJQaAqKqaOXEWzDRKTeBbx0ogU52YXIi8A6uL1gX
/7YwQtry0lr4hLSu9dH982CeqO0RqxPg8oIXX7rKPGLulPWwunNyXkt3IqqhEak29dtY6IVecmF2tkOvvi
/Ml4/FBZDqEmaquYQQbhU1mHLgK19ctUqXlBa1F4yef3hWXrWglD5Cdm+9pTRLhdD982CLuA2hr
/PhB5fSfv7+2FLtRjbgOjIDCznkM0nwgakWBd49GIFOdxFyIvfamiu54psh18DqovIbWYckeGEx6eeEriz4gU53gXIjwqp
/W7kCmyDwdNtmF+485PGLMbGoltHcv+70yteUd9eDqporuJKbIdQAq/MDCumUtK710B3zYawMPGVKBoRkQiEv/m/JZu/VIJQ
/fNgi7gNoa/z4QeX0n7+/thS7UY24XqScxsrdyDbsQH4h/FuDDxlSgaBc0LzqptLvOKaQdUU38YROgh0INtwwn0klA+MbfS+cwF3
/B0RX+phNOPRDE2iDwuZbUz2h
/ZkiXlRf2glq8Lwkm6qEwkC5Hlm68kCSrldQd9bWfvOz0y2QBpUaDPs6+vER+FJTk3qYSDj0RwV7wRZiS4Cp46ISOcmFz1sJ
/vq+JBqrWN3DrIsKaiIcjSPEBicYBjE1sVKpHtQRFE77vj8iG/qHQdNq2F+48EMROEFWccub/bOxl2xb2k3TCPk4
/CHKaQdT038cTntkAwc4UVL0S5ItZiIHalzVnlhNvv+/YV/6Rp3M4UsLa+EQwGfIziSZVb4z8NkmWtUdGEy
```

Figure 17 - Configuration sent by a Lumma C2 server.

Using a tool found on GitHub, we could decode the encrypted configuration. It contained 492 lines of JSON data:



```
{
    "v": 1,
    "c": [
        {
            "t": 0,
            "p": "%userprofile%",
            "m": "*.txt",
            "z": "Important Files/Profile",
            "d": 1
        },
        {
            "t": 0,
            "p": "%userprofile%",
            "m": "*key*",
            "z": "Important Files/Profile",
            "d": 1
        },
        {
            "t": 0,
            "p": "%userprofile%",
            "m": "*bitcoin*",
            "z": "Important Files/Profile",
            "d": 3
        },
        {
            "t": 0,
            "p": "%userprofile%",
            "m": "*binance*",
            "z": "Important Files/Profile",
            "d": 3
        },
```

Figure 18 - Beginning of the configuration file.

With the endpoint "**`/login`**", the operator has access to the C2 panel.

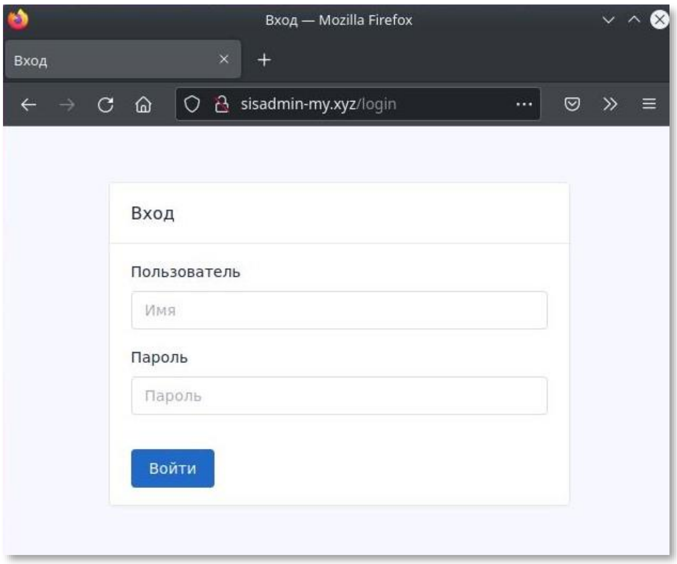
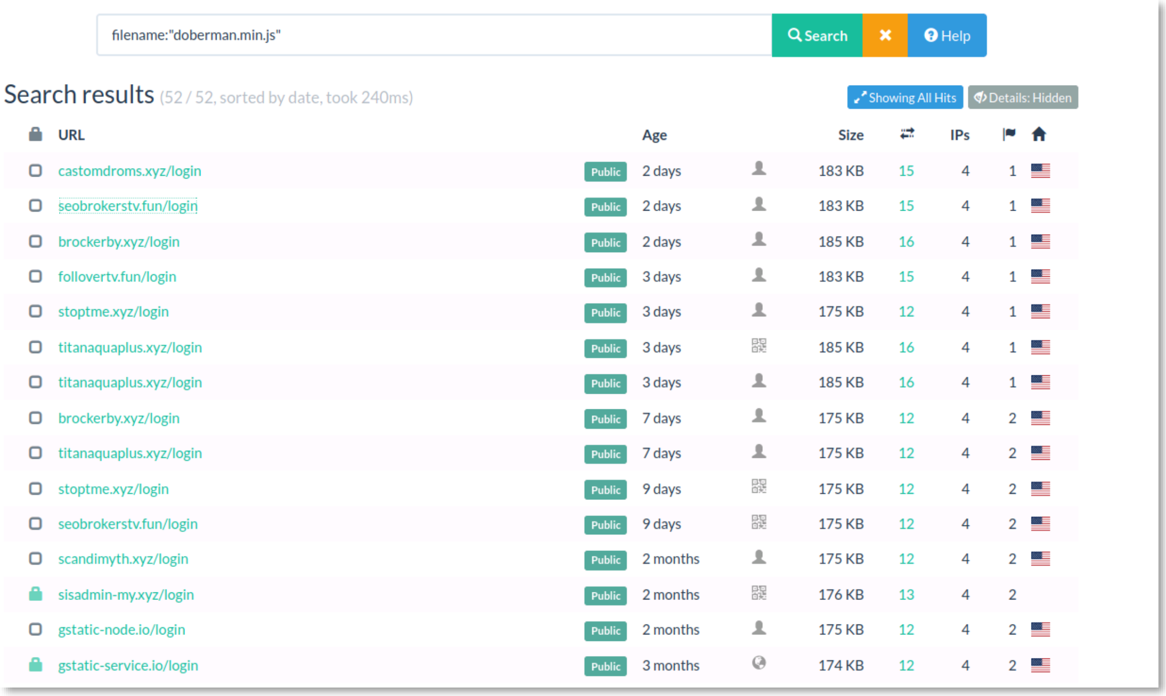


Figure 19 - Example of a Lumma C2 panel access.

Using the filename “**`doberman.min.js`**” which is found at this URL of Lumma panels “**`*/core/panel/js/doberman.min.js`**”, we can identify 29 other Lumma C2 panels:



filename:"doberman.min.js"

Search results (52 / 52, sorted by date, took 240ms)

🔒	URL		Age		Size	⇄	IPs	⚑	⌂
□	castomdroms.xyz/login	Public	2 days	👤	183 KB	15	4	1	🇺🇸
□	seobrokerstv.fun/login	Public	2 days	👤	183 KB	15	4	1	🇺🇸
□	brockerby.xyz/login	Public	2 days	👤	185 KB	16	4	1	🇺🇸
□	followertv.fun/login	Public	3 days	👤	183 KB	15	4	1	🇺🇸
□	stoptme.xyz/login	Public	3 days	👤	175 KB	12	4	1	🇺🇸
□	titanaquaplus.xyz/login	Public	3 days	▦	185 KB	16	4	1	🇺🇸
□	titanaquaplus.xyz/login	Public	3 days	👤	185 KB	16	4	1	🇺🇸
□	brockerby.xyz/login	Public	7 days	👤	175 KB	12	4	2	🇺🇸
□	titanaquaplus.xyz/login	Public	7 days	👤	175 KB	12	4	2	🇺🇸
□	stoptme.xyz/login	Public	9 days	▦	175 KB	12	4	2	🇺🇸
□	seobrokerstv.fun/login	Public	9 days	▦	175 KB	12	4	2	🇺🇸
□	scandimyth.xyz/login	Public	2 months	👤	175 KB	12	4	2	🇺🇸
🔒	sisadmin-my.xyz/login	Public	2 months	▦	176 KB	13	4	2	
□	gstatic-node.io/login	Public	2 months	👤	175 KB	12	4	2	🇺🇸
🔒	gstatic-service.io/login	Public	3 months	🌐	174 KB	12	4	2	🇺🇸

Figure 20 - Search results on Urlscan.io for the filename “doberman.min.js”.

As the filename is mentioned in the HTML “ **<head>**” section in this format: “***<script src="/core/panel/js/doberman.min.js"></script>***”, searching for it in online services search engine gives us respectively 19 and 12 additional C2. Once sorted, we obtain **41 unique Lumma C2 servers**. Five of them were active during the investigation. They all had the same configuration we already decrypted earlier.

3.2 New version of Lumma stealer

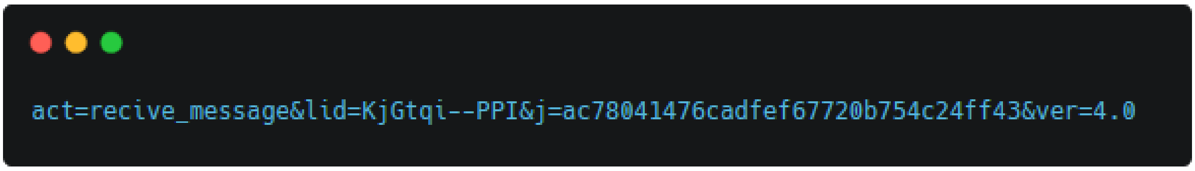
In the latest version of Lumma stealer, communications with the C2 server no longer rely on GET requests, but only on POST requests[1]. The former endpoints were also replaced by a unique one (“`/api`”) accepting different parameters:

- **`act=recive_message`** for retrieving the C2 configuration
- **`act=send_message`** for data exfiltration

The payload must also send three other parameters to get the configuration data:

- **`lid`** (meaning “Lumma ID”)
- **`j`** (Victim ID ??)
- **`ver=`** (Potentially the Lumma version number)

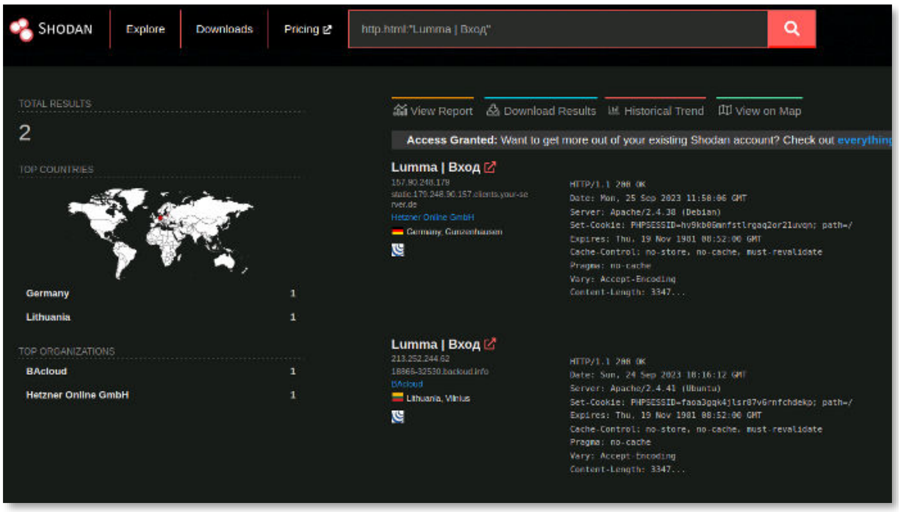
[1] https://x.com/g0njxa/status/1702444978503360989?s=20



```
act=recive_message&lid=KjGtqi--PPI&j=ac78041476cadfef67720b754c24ff43&ver=4.0
```

Figure 21 - Parameters sent to the C2 domain dedoxtrone[.]fun.

We also noticed the absence of the file "**doberman.min.js**" on new Lumma C2 panel, while the file "**dober.css**", also found on older version of the panel, is still there and that the html title has changed from "**Вход**" (meaning *login*) to "**Lumma | Вход**". Searching for this html title on Shodan gives us additional C2 :



Access Granted: Want to get more out of your existing Shodan account? Check out everything

Lumma | Вход
157.90.248.179
static.179.248.90.157.clients.your-server.de
Hetzner Online GmbH
Germany, Gunzenhausen

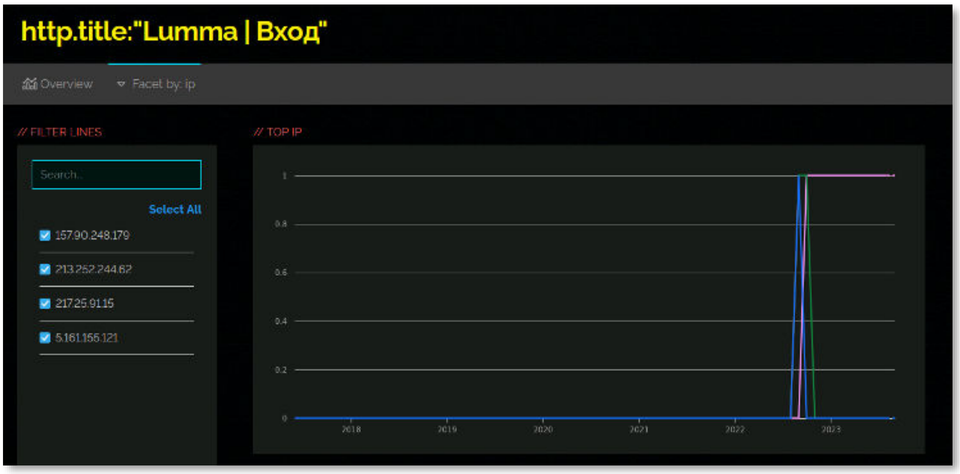
```
HTTP/1.1 200 OK
Date: Mon, 25 Sep 2023 11:58:06 GMT
Server: Apache/2.4.38 (Debian)
Set-Cookie: PHPSESSID=hv9kb06mnfstlrgaq2or2luvqn; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 3347...
```

Lumma | Вход
213.252.244.62
18866-32530.bacloud.info
BAcloud
Lithuania, Vilnius

```
HTTP/1.1 200 OK
Date: Sun, 24 Sep 2023 18:16:12 GMT
Server: Apache/2.4.41 (Ubuntu)
Set-Cookie: PHPSESSID=faoa3gqk4jlsr87v6rnfchdekp; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 3347...
```

Figure 22 : Shodan results for the query: http.title:"Lumma | Вход" showing 2 additional C2 IP addresses https://www.shodan.io/search?query=http.html%3A%22Lumma+%7C+%D0%92%D1%85%D0%BE%D0%B4%22.

Using shodan's trend feature, we find 2 additional IP addresses:



☑ 157.90.248.179
☑ 213.252.244.62
☑ 217.25.91.15
☑ 5.161.155.121

Figure 23: Shodan Trends results for the query: http.title:"Lumma | Вход" showing 2 additional C2 IP addresses. |https://trends.shodan.io/search?query=http.title%3A%22Lumma+%7C+%D0%92%D1%85%D0%BE%D0%B4%22#facet/ip.

III - Actionable content

1. IoCs

Value	Type	Description
95.163.238[.]17	IP	Lumma[.]site
lumma[.]**site**	Domain	Lumma shop
lumma[.]**online**	Domain	Lumma shop
astrolco[.]fun	domain	Lumma C2 (new version)
dedoxtrone[.]fun	domain	Lumma C2 (new version)
doggyguffy[.]fun	domain	Lumma C2 (new version)
fartyfun[.]fun	domain	Lumma C2 (new version)
fireworld[.]fun	domain	Lumma C2 (new version)
funnycox[.]fun	domain	Lumma C2 (new version)
markuschop[.]fun	domain	Lumma C2 (new version)
shoppervik[.]fun	domain	Lumma C2 (new version)
treepledeeple[.]fun	domain	Lumma C2 (new version)
Farformator[.]fun	domain	Lumma C2 (new version)
liveswords[.]xyz	domain	Lumma C2 (old version) Deployed by SocGholish
109.105.198[.]114	IPv4 address	Lumma C2 (old version)
144.76.173[.]247	IPv4 address	Lumma C2 (old version)
168.119.4[.]83	IPv4 address	Lumma C2 (old version)
170.130.55[.]129	IPv4 address	Lumma C2 (old version)
185.99.132[.]51	IPv4 address	Lumma C2 (old version)
185.99.133[.]246	IPv4 address	Lumma C2 (old version)
192.236.233[.]253	IPv4 address	Lumma C2 (old version)
195.123.227[.]138	IPv4 address	Lumma C2 (old version)
209.141.42[.]26	IPv4 address	Lumma C2 (old version)
212.113.116[.]46	IPv4 address	Lumma C2 (old version)
217.12.206[.]197	IPv4 address	Lumma C2 (old version)
217.25.91[.]15	IPv4 address	Lumma C2 (old version)
23.254.225[.]133	IPv4 address	Lumma C2 (old version)
5.161.155[.]121	IPv4 address	Lumma C2 (old version)
45.15.25[.]190	IPv4 address	Lumma C2 (old version)
45.8.146[.]130	IPv4 address	Lumma C2 (old version)
45.8.146[.]213	IPv4 address	Lumma C2 (old version) BlueCharlie Infrastructure
45.8.146[.]227	IPv4 address	Lumma C2 (old version) BlueCharlie Infrastructure
45.9.74[.]78	IPv4 address	Lumma C2 (old version)
77.73.134[.]51	IPv4 address	Lumma C2 (old version)

77.73.134[.]68	IPv4 address	Lumma C2 (old version)
78.46.190[.]160	IPv4 address	Lumma C2 (old version)
82.117.255[.]127	IPv4 address	Lumma C2 (old version)
82.117.255[.]128	IPv4 address	Lumma C2 (old version)
82.117.255[.]80	IPv4 address	Lumma C2 (old version)
82.118.23[.]50	IPv4 address	Lumma C2 (old version)
85.239.62[.]218	IPv4 address	Lumma C2 (old version)
94.142.138[.]26	IPv4 address	Lumma C2 (old version)
94.158.244[.]69	IPv4 address	Lumma C2 (old version)
213.252.244[.]62	IPv4 address	Lumma C2 (new version)
188.114.97[.]3	IPv4 address	Lumma C2 (new version)
157.90.248[.]179	IPv4 address	Lumma C2 (new version)
brockerby[.]xyz	domain	Lumma C2 (old version)
castomdroms[.]xyz	domain	Lumma C2 (old version)
cs6.csserv[.]ru	domain	Lumma C2 (old version)
follovertv[.]fun	domain	Lumma C2 (old version)
gstatic-node[.]io	domain	Lumma C2 (old version)
gstatic-service[.]io	domain	Lumma C2 (old version)
oneinformationcrypto[.]com	domain	Lumma C2 (old version) BlueCharlie infrastructure
scandimyth[.]xyz	domain	Lumma C2 (old version)
seobrokerstv[.]fun	domain	Lumma C2 (old version)
sisadmin-my[.]xyz	domain	Lumma C2 (old version)
stateinfospace[.]com	domain	Lumma C2 (old version) BlueCharlie infrastructure
static.247.173.76.144.clients.your-serve[.]de	domain	Lumma C2 (old version)
stoptme[.]xyz	domain	Lumma C2 (old version)
titanaquaplus[.]xyz	domain	Lumma C2 (old version)
static.121.155.161.5.clients.your-server[.]de	domain	Lumma C2 (old version)
evetesttech[.]net	domain	Lumma C2 (old version)
astrolco[.]fun	domain	Lumma C2 (new version)
coldwinded[.]fun	domain	Lumma C2 (new version)
18866-32530.bacloud[.]info	domain	Lumma C2 (new version)
static.179.248.90.157.clients.your-server[.]de	domain	Lumma C2 (new version)
33202cbfa6e4767b63f4dd9eb5b653d32761aa80f0ee11eaba822e0f444f3ad7	SHA-256	"Business License#60674.html"
5798de760b193a06cd9cb1c197feae081e2cff84ba5514a53ad313341b95663e	SHA-256	"Balance Sheet#37553.html"
5e4e078a71f6247b9e7c4569ec90a7ddec5f7bece465fc0c177587d920cd5aac	SHA-256	"Employee Contract_14212.html"
772d84d31e0a10a6dc6b7bb9bc2f71e135260e95a79e18b2145d53f678639232	SHA-256	"Request for Proposal (RFP)#51982.html"

e58a6c6ab2fa3d5e7ea3f13421f7818d614051e3c8d8cf360c3192c82df6a508	SHA-256	Lumma – Executable – Deployed by SocGholish
dd5b52a63e8a774c058e558aa7e983d6aa51f560ba3f01829287c4b85081b884	SHA-256	Lumma – Executable
587c5f89248035fe563f06a8b991f081418e6013cc0e77f9e052de59b758c1c1	SHA-256	dober.css
1377f22be21e2ae441b97eb6323198f963cfa0e246de83f00631cf6e9ba279e8	SHA-256	doberman.min.js

2. Recommendations

- Cookie-stealing malware incidents should be tackled seriously. Accounts should be reset, and browser caches should also be cleared.
- Consider a proactive employee’s credential assessment (logs, session cookies, login/pass etc.) on prioritized Dark web forums by CTI teams to mitigate the risk of account takeover.
- Block indicators of compromise included in the IoCs section of this report.

3. Sources

➢ https://www.esentire.com/blog/the-case-of-lummac2-v4-0
➢ https://fr.darktrace.com/blog/the-rise-of-the-lumma-info-stealer
➢ https://malpedia.caad.fkie.fraunhofer.de/details/win.lumma
➢ https://www.rapid7.com/blog/post/2023/08/31/fake-update-utilizes-new-idat-loader-to-execute-stealc-and-lumma-infostealers/
➢ https://cyble.com/blog/lummac2-stealer-a-potent-threat-to-crypto-users/
➢ https://twitter.com/sekoia_io/status/1572889505497223169
➢ https://cyble.com/blog/lummac-stealer-leveraging-amadey-bot-to-deploy-sectoprat/
➢ https://0xtoxin.github.io/malware%20analysis/Lumma-Breakdown/