PrivateLoader: the loader of the prevalent ruzki PPI service

By Pierre Le Bourhis, Quentin Bourgue and Sekoia TDR

Published: 2022-09-15 · Archived: 2026-04-05 18:30:49 UTC

Table of contents

- Background
- Introduction
- PrivateLoader, a widespread loader in the PPI landscape
- Technical analysis
  - Communications
  - PrivateLoader infrastructure
- Malware distributed by PrivateLoader
- Ruzki, a popular Pay-Per-Install service
- What is the ruzki Pay-per-Install service?
- PrivateLoader’s association with the ruzki Pay-per-Install service
  - Statistic URLs of PrivateLoader in publications of ruzki service
  - Redline campaigns associated to Ruzki botnets
  - A perfect timing
- Conclusion
- MITRE ATT&CK TTPs
- Annex
- IoCs

Background

Pay-Per-Install (PPI) is a **malware service** widely used in the cybercrime ecosystem that **monetises the installation of malicious software**. As generally observed, a malware operator provides a Pay-per-Install service operator with a payload, a requested number of installations, and a target geographical location. The service operators are then responsible for distributing the malware sample based on the customer’s request, and for which they will be paid. Actors selling “installs” play a **key role in the distribution of threats**, as well as the underground economy.

To run this service, Pay-per-Install operators use tools to monitor the number of installations, location of infected hosts, and other valuable installations-related information. **For the most premium services**, actors also operate a **modular loader enabling management of additional payloads**, customising the attack chain, improving the rate of successful compromise by reducing the detection of payloads.

While most Pay-per-Install services use their own traffic distribution network, some purchase traffic generation services, such as those offered by traffers teams, to ensure wide and fast distribution of malware samples.

Introduction

Sekoia.io observed that PrivateLoader is one of the **most widely used loaders in 2022**. It is used by a Pay-Per-Install service to deploy multiple malicious payloads on the infected hosts.

First observed in May 2021, PrivateLoader is a **modular malware whose main capability is to download and execute** one or several payloads. The loader implements anti-analysis techniques, fingerprints the compromised host and reports statistics to its C2 server.

SEKOIA analysts tracked PrivateLoader's network infrastructure for several months and recently conducted an in-depth analysis of the malware. In parallel, we also monitored activities related to the *ruzki* PPI malware service.

The threat actor *ruzki* (aka *les0k*, *zhigalsz*) advertises their PPI service on underground Russian-speaking forums and their Telegram channels under the name *ruzki* or *zhigalsz* since at least May 2021. Their business model consists in **selling bundles of thousand installations**, located on systems all over the world, or specifically in Europe or in the United States.

Our **investigations on Dark Web** forums allow us to assess with high confidence that **PrivateLoader is the proprietary loader of the *ruzki* PPI malware service**.

PrivateLoader, a widespread loader in the PPI landscape

PrivateLoader is a downloader malware family first publicly reported in February 2022 by Intel471. The loader is used as part of a PPI service, to deliver payloads of multiple malware families operated by several threat actors or intrusion sets.

Following is an overview of the PrivateLoader malware capabilities, infrastructure and malware involved in its campaigns.

Technical analysis

PrivateLoader is a modular C++ loader composed of three modules, including the loader to load the Core module, the Core module contacting the Command and Control (C2) to get the URL to download the next payload, and the Service module ensuring persistence.

PrivateLoader's main purpose is to provide an environment where the next payload is downloaded and executed. To be agnostic, the malware does not embed the next stage, the loader downloads the next payload based on its configuration. The downloaded payload is obfuscated by customised operations developed by the author and detailed later.

PrivateLoader core module offers the following functionalities:

- Stack string obfuscation;
- Host fingerprint (used for victim statistic on the C2 panel);
- Next stage payload download over HTTPS and execution;
- Anti-analysis techniques.

Execution of some samples highlight the presence of specifics Tactics, Techniques and Procedures (TTPs) [https://tria.ge/220717-gpte2ahcbp, https://tria.ge/220826-kv1m3saahk]:

- Impair defenses (disables Windows Defender Real Time protection);

The loader has its own Command and Control, used to gather victim statistics and to send next stage URLs via multiple dead drop resolvers.

PrivateLoader's main use consists in loading one or several third party malware. To do so, the loader contacts hardcoded URLs which are obfuscated in the PE, it then requests the URL(s) that returns a dead drop resolver, subsequently used to get another URL hosting the next stage payload.

After downloading the next stage, PrivateLoader deobfuscates its content using byte replacement based on the following table:

Original Byte	Replacement byte
0x00	0x80
0x80	0x0a
0x0a	0x01
0x01	0x05
0x05	0xde
0xde	0xfd
0xfd	0xff
0xff	0x55
0x55	0x00

Table 1 : Byte replacement table

After the bytes substitution, the loader xors the payload content with the key `0x9d`. The downloaded deobfuscated file starts with a magic that is removed before executing the valid PE. A script to extract obfuscated payload is provided in the annex.

Communications

The communication of PrivateLoader is split in three parts:

1. The malware deobfuscates one or several embedded URL(s) not controlled by the attacker (tactic of dead drop resolver);
2. It requests the embedded URL, the response is in plaintext and follows this format: `HOST:<C2 IP address>`;
3. The infected host then requests the C2 over HTTP on a unique endpoint `/base/api/statistics.php`, the response contains the final payload URL obfuscated using XOR operation;
4. Eventually, the malware downloads the next payload again, obfuscated with a different algorithm.

No.	Time	Source	Destination	Protocol	Length	Info
36	52.8027320…	10.127.0.125	172.67.34.170	HTTP	283	GET /raw/A7dSG1te HTTP/1.1
39	52.8123490…	172.67.34.170	10.127.0.125	HTTP	91	HTTP/1.1 200 OK (text/plain)
44	52.8290810…	10.127.0.125	212.193.30.115	HTTP	269	GET /base/api/statistics.php HTTP/1.1
45	52.8345430…	212.193.30.115	10.127.0.125	HTTP	358	HTTP/1.1 200 OK (text/html)
60	60.9547660…	10.127.0.125	107.182.129.2…	HTTP	290	HEAD /download/PL_Client.bmp HTTP/1.1
66	65.2516000…	10.127.0.125	107.182.129.2…	HTTP	270	GET /download/PL_Client.bmp HTTP/1.1

Figure 1. PrivateLoader communication overview

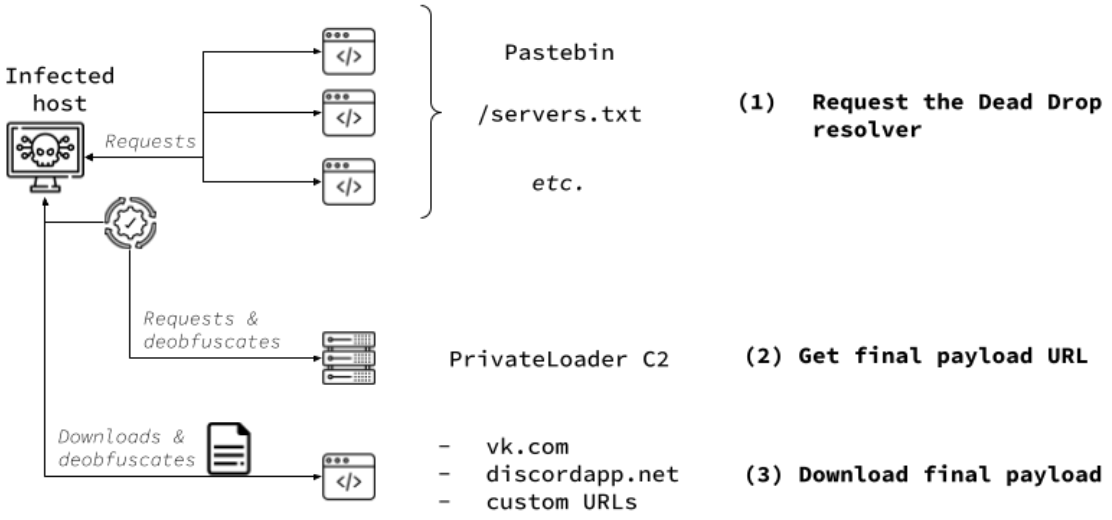


Image 2: PrivateLoader core module network communication overview

While the Dead Drop Resolver technique used by PrivateLoader can leverage legitimate websites such as Pastebin, the malware can also use proprietary servers with URLs like `/servers.txt` and `/proxies.txt`. In the latter Dead Drop Resolver option, C2 IP is drowned in a list of IP addresses, the correct one is obfuscated by scrambling.

The communication with PrivateLoader C2 is obfuscated xoring the HTTP body with the key `0x6d`, a technique consistently observed across all campaigns.

SEKOIA observed that PrivateLoader operators changed their final payload hosting provider over the summer, shifting from Discord attachments to vk.com documents to host the downloaded payload. It is possible that PrivateLoader customers are able to provide their own server to host the payload. As increasingly used by several malware as C2 or as a files hoster, Discord is now under more scrutiny. It is possible this increased monitoring was a driver for PrivateLoader’s recent hosting shift.

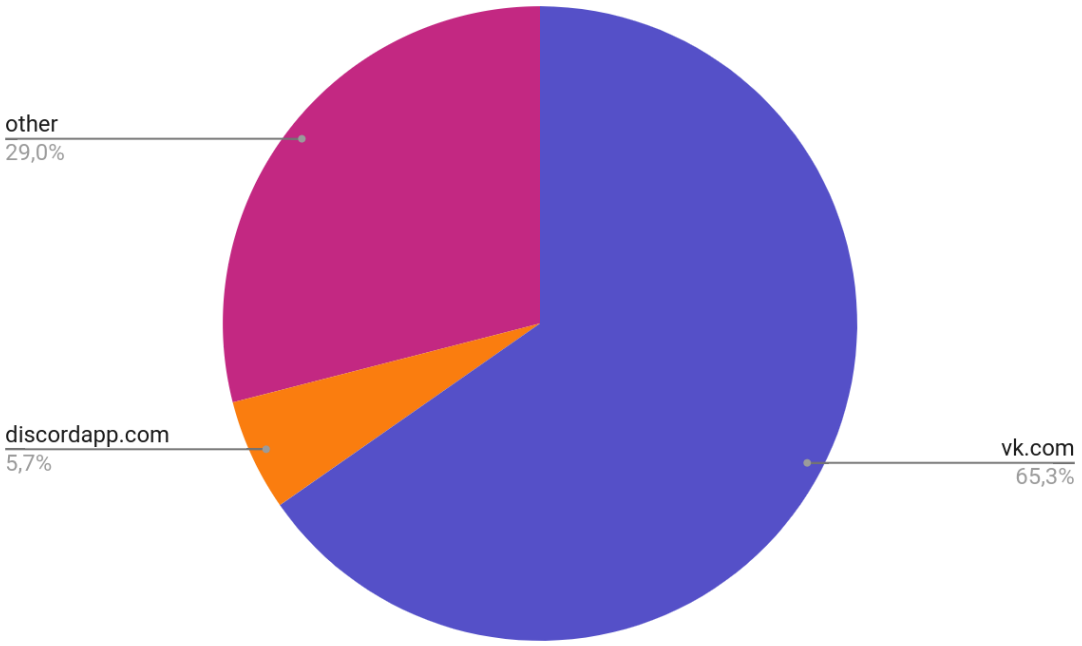


Figure 3. Domains hosting final payload downloaded by PrivateLoader observed in between 15 August and 3 September 2022

After downloading the next stage, if the core module is not configured to set up the service module, communication with the PrivateLoader Core module is cancelled. Core module allows two configurations, the first one will contact the C2 to load the Service module, the second configuration downloads other malwares and ends its activity, all remaining communication in this specific configuration belongs to the next stage. SEKOIA observed the second behaviour as the most widespread among campaigns involving PrivateLoader malware.

PrivateLoader infrastructure

The infrastructure hosts the C2 servers used by PrivateLoader operators to manage their service. PrivateLoader samples communicate with these C2 servers to retrieve and exfiltrate data, as previously detailed in the analysis.

At the time of writing, SEKOIA observed 4 currently active C2 servers:

IP address	ASN	Country	Active since
79.174.12.174	RU-JSCIOT (29182)	Russia	2021-06-30
91.240.85.160	RU-JSCIOT (29182)	Russia	2021-08-31
212.193.30.115	AS_DELIS (211252)	Czechia	2022-07-26
167.235.29.244	HETZNER-AS (24940)	Germany	2022-09-04

Table 2. Active C2 servers on 6 September, 2022

Requests on publicly exposed endpoints (see the Part “Statistic URLs of PrivateLoader in publications of ruzki service”) suggest that all IP addresses redirect to the same C2. This information is valuable in associating the exclusive use of PrivateLoader to a specific PPI service.

To proactively track the PrivateLoader servers, we identified a heuristic based on the characteristic HTTP response and the HTTP headers on port 80. The servers respond in 200 with a HTML page entitled "*404 not found*" containing "*(Ubuntu) Server at*", with PHP server headers.

SEKOIA also track PrivateLoader servers by searching for the specific PrivateLoader URLs (*/base/api/getData.php* and */base/api/statistics.php*) and their administration panel.

Monitoring the PrivateLoader infrastructure, we observed over 30 unique C2 servers. The volume of malicious traffic generated by these servers suggests that the servers are fastly detected by security vendors. Moreover, it is highly likely that the servers are eventually shut down if they are not provided by a bulletproof host.

Malware distributed by PrivateLoader

In last weeks' campaigns SEKOIA observed, the following malware families were actively distributed by PrivateLoader payloads:

- Information stealers: Redline, Vidar, Raccoon, Eternity, Socelars, Fabookie, YTStealer, AgentTesla, Phoenix and other uncategorized stealers.
- Ransomware: Djvu.
- Botnet: Danabot, SmokeLoader.
- Miners: XMrig and other uncategorized stealers.
- Other commodity malware: DcRAT, Glupteba, Netsupport, and Nymaim variant.

PrivateLoader distribution of malware families is documented in Intel471 and Bitsight reports, respectively over the course of September-January 2021, and July-August 2022.

Based on the **wide selection of malware families**, which implies a **wide range of threat actors or intrusion sets operating this malware**, the **PPI service running PrivateLoader is very attractive and popular to attackers** on underground markets.

Ruzki, a popular Pay-Per-Install service

SEKOIA observed that PrivateLoader is **one of the most widespread loaders** in 2022. To better understand the spread of PrivateLoader, we must consider the PPI service associated with the loader.

SEKOIA **investigations on Dark Web forums** led us to take a closer **look at the *ruzki* PPI malware service**. Our observations from *ruzki* customer publications, botnets delivered by PrivateLoader and *ruzki* messages allowed us to find evidence to link this PPI service and the loader.

After a brief description of the *ruzki* service, we present how we have **associated this PPI service to the PrivateLoader malware**.

What is the *ruzki* Pay-per-Install service?

The *ruzki* PPI service consists of selling bundles of a thousand installations located on systems all over the world, or especially in Europe or in the United States.

Since May 2021, the *ruzki* (aka *les0k*, *zhigalsz*) profile advertised the PPI service under the same name on the Lolz Guru cybercrime forum. Their publications include pricing information, number of installs per day the service can afford for

one customer, and the traffic source. Similar publications are frequently posted by the same profile on the Telegram channel (t[.]me/ZHIGALSZinstalls) associated with the services.

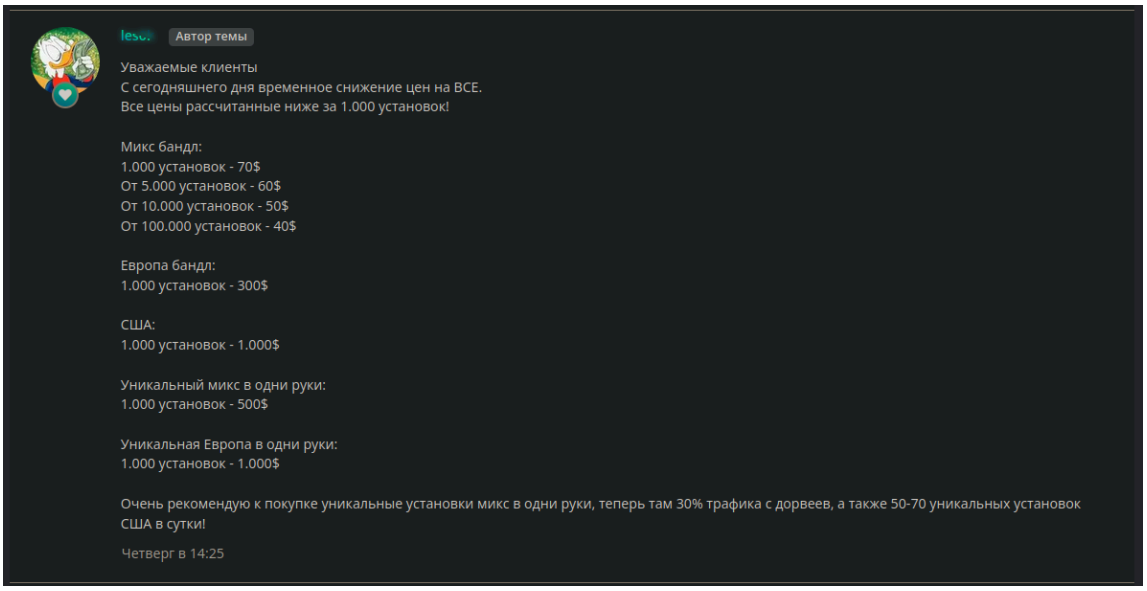


Figure 4. Les0k's publication related to ruski's PPI service pricing on 1 September, 2022 (source: Lolz Guru)

On 1 September, 2022, buying 1,000 installations on infected systems:

- Worldwide (also called *Mix world installs*) costs $70;
- Located in Europe costs $300;
- Located in the United States costs $1,000.

All prices are degressive based on the quantity.

Ruzki threat actor recommends customers to buy *unique installs* packages for higher output. Indeed, a single classic installation can be sold to several customers, making the output of the compromise low for them. While *unique installs* are sold to a single customer. The exploitation of these installations is therefore more profitable since the customer has exclusive access.

At its launch, the service could offer up to 20,000 installations per day. As of today, we have not found numbers on the current capabilities of the PPI service. The traffic generation relies on an affiliate network that represented, in May 2021, 800 *webmasters* leveraging multiple infection chains. SEKOIA suspects that one or more traffers teams are behind these *webmasters*.

SEKOIA analysts were able to establish ruzki's PPI *modus operandi* based on customers' feedback, notably shared through screenshots of their conversations with the *ruzki* PPI service operator. On another note, most of the feedback was written following the ruzki operator on their Telegram channel.

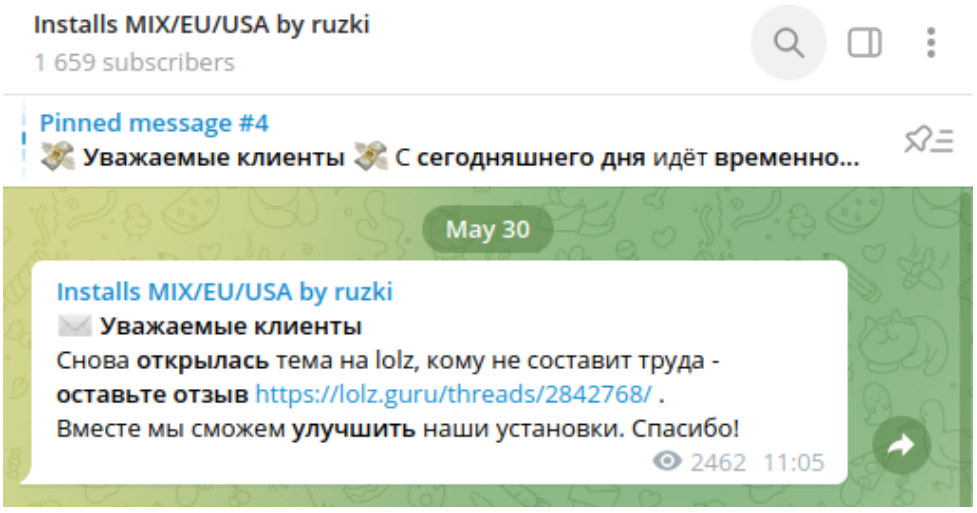


Figure 5. Ruzki service operator post requesting feedback in their Telegram channel 30 mai

(traduction du message)

Below is an outline of the service, as observed by SEKOIA:

Step 1: The malware operator willing to distribute its payload contacts the *ruzki* service's operator on Telegram (t[.]me/zhigalsz), providing its requested number of installs and the target geographical location.

Step 2: The PPI service's operator sets the price of the request, and provides the malware operator with a cryptocurrency wallet address.

Step 3: The malware operator provides a proof of the transfer (oftentimes a screenshot of a cryptocurrency application) to conclude the financial transaction, and supplies the payload it wants to distribute.

Step 4: The *ruski* service's operator shares a password-protected link to follow statistics on the number of installs related to the customer's payload.
Step 5: *Webmasters* are responsible for distributing the customer payload.

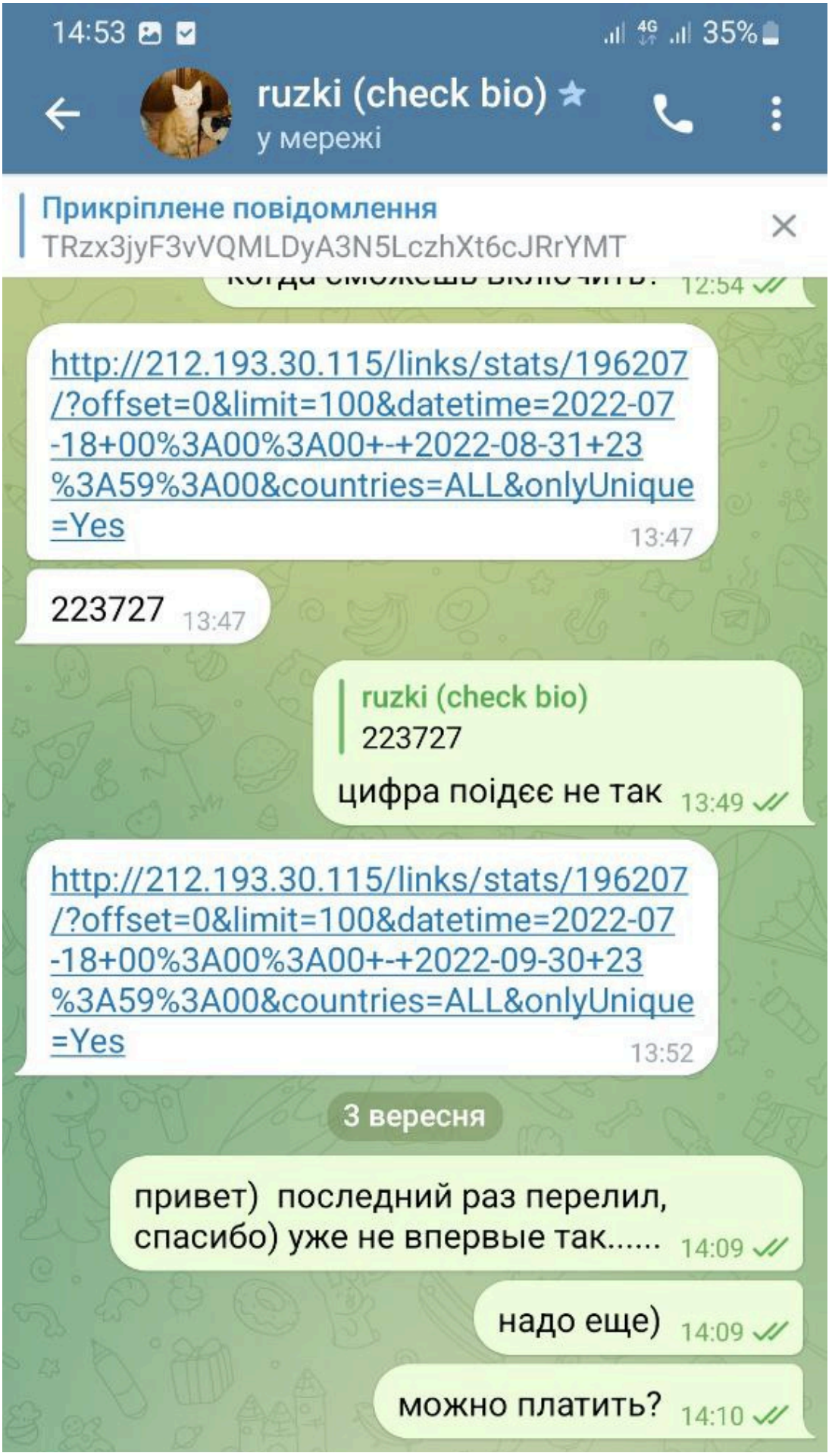
Further analysis of the *ruzki* PPI service allowed SEKOIA to establish a link with PrivateLoader.

PrivateLoader's association with the *ruzki* Pay-per-Install service

Several observations allow us to associate with a high level of confidence the PrivateLoader as the main tool of the ruzki PPI service.

Statistic URLs of PrivateLoader in publications of *ruzki* service

In recent months, malware operators who subscribed to the *ruzki* service shared screenshots of conversation with *ruzki*. These conversations contain a URL provided by the PPI service operator to its customers to monitor statistics related to their campaign of installations.


14:53
35%
ruzki (check bio)
у мережі
Прикріплене повідомлення
TRzx3jyF3vVQMLDyA3N5LczhXt6cJRrYMT
12:54
http://212.193.30.115/links/stats/196207/?offset=0&limit=100&datetime=2022-07-18+00%3A00%3A00+-+2022-08-31+23%3A59%3A00&countries=ALL&onlyUnique=Yes
13:47
223727
13:47
ruzki (check bio)
223727
цифра поідєє не так
13:49
http://212.193.30.115/links/stats/196207/?offset=0&limit=100&datetime=2022-07-18+00%3A00%3A00+-+2022-09-30+23%3A59%3A00&countries=ALL&onlyUnique=Yes
13:52
3 вересня
привет) последний раз перелил, спасибо) уже не впервые так.....
14:09
надо еще)
14:09
можно платить?
14:10

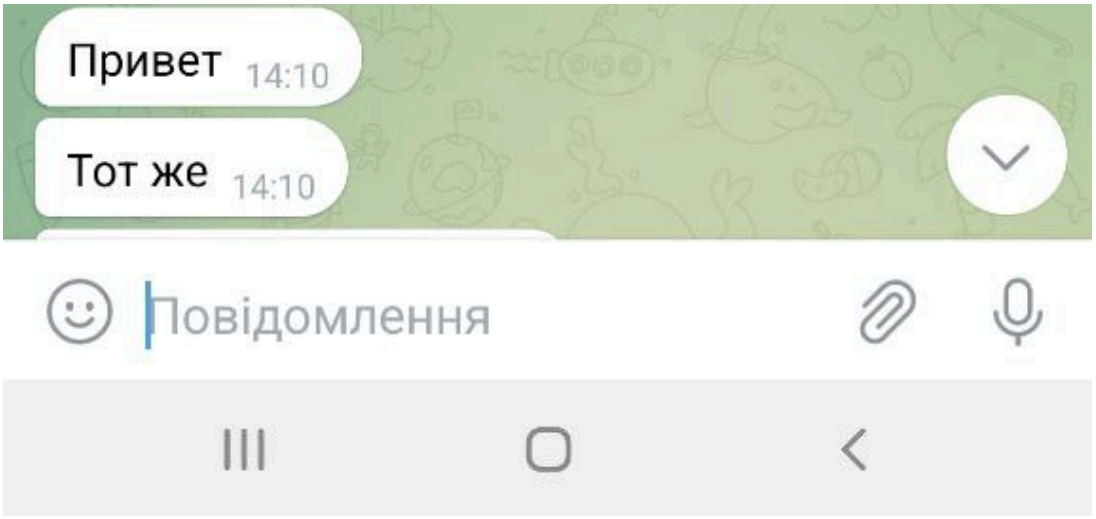


Figure 6. Capture of the conversation between a customer of the ruzki PPI service and ruzki (source: Lolz Guru)

These URLs match those of the PrivateLoader C2 servers and the endpoint */links/stats* matches that of a PrivateLoader C2 server.

IP addresses mentioned by *ruzki* customers	**PrivateLoader C2 in SEKOIA.IO CTI**
45.144.225[.]243 on May 30, 2022	Active since 2021, until August 26, 2022
212.193.30[.]115 on August 24, 2022	Active since July 17, 2022, and still active at moment of writing

Table 3. IP addresses mentioned by *ruzki* PPI customers and retrieved via SEKOIA C2 Trackers

Redline campaigns associated to *Ruzki* botnets

While tracking PrivateLoader, we retrieved multiple PrivateLoader samples downloading the Redline information stealer as a final payload. We were able to associate a majority of retrieved Redline's samples to PrivateLoader-loaded botnets, including *ruzki9, nam6.2, cryptex, @forceddd_lzt, ruzki, don_karl_installs, 3108_RUZKI, persom, 2007329039* and *riii_ff*.

While evidence is very thin at this stage to formally associate these Redline botnets to ruzki, several botnets' names suggest some sort of connection to the threat actor. One hypothesis is that ruzki's customers could name their botnet based on the PPI service they use to distribute their samples. Another hypothesis is that ruzki operates the Redline malware in parallel to the PPI service activity.

Regardless of the operators behind the Redline samples, it is yet another indication of a likely connection between PrivateLoader and ruzki.

A perfect timing

The *ruzki* operator was active on Telegram and Russian-speaking underground forums since at least May 2021:

- The profile *les0k* advertising the *ruzki* PPI service joined Lolz Guru on May 1, 2021;
- zhigalsz, the first Telegram channel associated with the service was created on May 7, 2021.

Additionally, Intel471 assesses that PrivateLoader is used since at least May 2021. Building upon previously outlined observations, this increases our confidence in associating PrivateLoader with the *ruzki* PPI service.

In addition to the URLs shared to *ruzki* customers and Redline botnets named using “ruzki”, we observed that the *ruzki* operator uses the term “our loader” in Russian on its Telegram channel.

Ruzki’s publication in Telegram channel associated with the service, on 16 February, 2022:

У нас своя сеть вебмастеров и мы являемсяклиенты, хочу вам соотрафика. Также наш лоадер каждый день чистится и отстук около 80%. (Translated from Russian: We have our own network of webmasters and we are clients, I want you to cooperate. Also, our loader is cleaned every day and the turnover is around 80%.)

Moreover, we identified a single botnet associated with all the PrivateLoader C2 servers. SEKOIA therefore assess with high confidence that **PrivateLoader is the proprietary loader of the *ruzki* PPI malware service**.

The **spread of PrivateLoader** could therefore **be explained by the following elements**:

- The popularity of the associated service among threat actors (competitive pricing, good service and support, accessibility to less experienced threat actors and intrusion sets);
- A distribution of PrivateLoader oriented on the quantity of installations, rather than on the quality;
- A good coverage of this threat within SEKOIA, and more generally within the cybersecurity community.

Conclusion

Pay-per-Install services always played a **key role in the distribution of commodity malware**, and more generally in the **increase of the threat surface exposure**. While the cybercrime ecosystem related to PPI services is constantly evolving with new threat actors and emerging malware, the ***ruzki* Pay-per-Install service using PrivateLoader is established for over a year**.

PrivateLoader became one of the most widespread loaders used for a PPI service in 2022. This downloader malware is used to deliver multiple malware including information stealers, ransomware, botnets and miners. Tracking PrivateLoader provides an **interesting insight into prevalent commodity malware in the cybercriminal landscape**, as well as uncovering unidentified and emerging malware. As yet another turnkey solution **lowering the cost of entry** into the cybercriminal market and a **service contributing to a continuous professionalisation** of the cybercriminal ecosystem, it is highly likely more PrivacyLoader-related activity will be observed in the short term.
To provide our customers with actionable intelligence, SEKOIA analysts will continue to track PrivateLoader C2 infrastructure, analyse malware technical evolution and monitor the *ruzki* Pay-per-Install service and their customers.

MITRE ATT&CK TTPs

Tactic	Technique
Command and Control	T1001 – Data Obfuscation
Command and Control	T1071.001 – Application Layer Protocol: Web Protocols
Command and Control	T1102.001 – Web Service: Dead Drop Resolver
Command and Control	T1105 – Ingress Tool Transfer

Command and Control	T1132 – Data Encoding
Command and Control	T1568 – Dynamic Resolution
Command and Control	T1573.001 – Encrypted Channel: Symmetric Cryptography
Defence Evasion	T1027 – Obfuscated Files or Information
Resource Development	T1608.001 – Stage Capabilities: Upload Malware

Table 4: Mitre Att&ck TTPs

Annex

Python script to deobfuscate payload dropped by PrivateLoader:

```
 import sys
from copy import copy

def deobfuscate(filename: str) -> None:

    print(f"deobfuscates privateloader file: `{filename}`")
    with open(filename, "rb" )as f:
        data = bytearray(f.read())

    data2 = copy(data)
    data2 = replace_all(data, data2, 0x00, 0x80)
    data2 = replace_all(data, data2, 0x80, 0x0a)
    data2 = replace_all(data, data2, 0x0a, 0x01)
    data2 = replace_all(data, data2, 0x01, 0x05)
    data2 = replace_all(data, data2, 0x05, 0xde)
    data2 = replace_all(data, data2, 0xde, 0xfd)
    data2 = replace_all(data, data2, 0xfd, 0xff)
    data2 = replace_all(data, data2, 0xff, 0x55)
    data2 = replace_all(data, data2, 0x55, 0x00)

    unxored = bytearray()
    for byte in data2:
        unxored.append(byte ^ 0x9d)

    with open(f"unxored-{filename}", "wb") as f:
        f.write(unxored[4:])

    print(f"unxored privateloader payload dumped in `unxored-{filename}` file")


def replace_all(data: bytearray, data2: bytearray, x: int, y: int) -> bytearray:

    print(f"replace all {hex(x)} by {hex(y)}")
```

```
    for index, byte in enumerate(copy(data)):
        if byte == x:
            data2[index] = y
    return data2

if __name__ == "__main__":
    deobfuscate(sys.argv[1])
```

IoCs

The list of IoCs is available on SEKOIA github repository.

IOC	Context	Link
hxxp://212.193.30[.]115/base/api/getData.php	PrivateLoader C2	app.sekoia.io
hxxp://212.193.30[.]115/base/api/statistics[.]php	PrivateLoader C2	app.sekoia.io
hxxp://116.203.105[.]117/base/api/getData.php	PrivateLoader C2	app.sekoia.io
hxxp://212.193.30[.]115/service/communication.php	PrivateLoader C2	app.sekoia.io
hxxps://pastebin[.]com/raw/A7dSG1te	PrivateLoader Dead Drop Resolver	app.sekoia.io
hxxp://163.123.143[.]4/proxies.txt	PrivateLoader Dead Drop Resolver	app.sekoia.io
hxxp://107.182.129[.]251/server.txt	PrivateLoader Dead Drop Resolver	app.sekoia.io
hxxp://wfsdragon[.]ru/api/setStats.php	PrivateLoader C2	app.sekoia.io

Table 5: PrivateLoader Infrastructure

IOC	Link
6c9223f75d2cca77fc09fbce2e76034326718c4daab02abc1e4f7caefefbcbc5	app.sekoia.io
2048e7a38a3f8b52bb3e47435ec8ed42dc531446af7a02f76a7f8f79665610de	app.sekoia.io
6aa0d341cee633c2783960687c79d951bf270924df527ac4a99b6bfabf28d4ae	app.sekoia.io
a0d021d03af4e6a87890bd0fb929e7f8ed83e08d73a0521c25957ad29cce2381	app.sekoia.io
0e14021b3594a5a54254d4f1cdf374dcf6650d71111f3dcf616f7043d7b2fba3	app.sekoia.io
e2c2d8bf5451525085df47bbb63776fffa381823cf591de29f8dfc692c36d42d	app.sekoia.io
21ce471527c051d26da04e96c2829f450b031767399ea401920ab8b43018e421	app.sekoia.io

Table 6: PrivateLoader hashes

IOC	Malware	Link
hxxp://linislominyt11[.]at	SmokeLoader	app.sekoia.io
hxxps://oshi[.]at/Kaqm	Agent Tesla	app.sekoia.io
hxxp://89.185.85[.]53/	Raccoon Stealer	app.sekoia.io
hxxp://146.70.87[.]133/	Raccoon Stealer	app.sekoia.io
5.182.36[.]101:31305	RedLine	app.sekoia.io
138.201.195[.]134:3202	RedLine	app.sekoia.io
109.107.181[.]244:41535	RedLine	app.sekoia.io
193.124.22[.]24:18114	RedLine	app.sekoia.io
193.124.22[.]24:18114	RedLine	app.sekoia.io
103.89.90[.]61:12036	RedLine	app.sekoia.io
185.215.113[.]55:1591	RedLine	app.sekoia.io
185.106.92[.]20:33168	RedLine	app.sekoia.io
107.189.31[.]171:80	Vidar	app.sekoia.io

Table 7: C2 of the payload dropped by PrivateLoader

IOC	Malware	Link
392049ce2edacaef91a29eb0ef2b7b9927a82550b592dedf725a33b6cfdd2381	DcRAT	app.sekoia.io
ff3ae8fff0d1862d4bde8f61e0ed14ef76d6d2cc6d940bb83dc0b4cfdacc2752	YTStealer	app.sekoia.io
456a46109fb5c42e3223592853934a52aa1cebeae6757e0e3792282c07750f32	NetSupport RAT	app.sekoia.io
866918dce85cab2200a0d271a8d6e7669296890d2d32ec3bea2fc78c6778a037	NetSupport RAT	app.sekoia.io
27d2943e3dc87f5bfaf314dbf2b50dad4563b53515d471f398b81d5fe8b7a8fe	RedLine	app.sekoia.io
ab0e35830bdaf3502d037d059b50f1e10c8283f5300565d6fb311d0827ac6ae8	RedLine	app.sekoia.io
88c7dbb90db43f552465fb2b3a2c036f5c906cf2c8f14b80ee3cab8eee52d31d	RedLine	app.sekoia.io
52651bd3091f375b41b38aeeffd45d4df8fe0b1763fb6788756b473e6f96b5e2	RedLine	app.sekoia.io
99d207e9df203331c4849506693c351f777ace02a0ddebce2e3296bd79d3b081	RedLine	app.sekoia.io
75c9e2a6c3d9196c4ea851f90401d6b9acae07489a41d462a462e42f26780215	RedLine	app.sekoia.io
202d14ca71ba0a0d0cd06d3bb0da7a4b74c5a3de429420d6c0a0b766b81cc4cc	RedLine	app.sekoia.io

de017a6129651d442c3e3c25c7f137d1da4264bd8cde6f67a7ed575d1001128a	RedLine	app.sekoia.io
be30847b4cf9553f18b98e00e5cdcbecf099cf0369a5f95ca1057b3f122f7185	RaccoonStealer	app.sekoia.io

External References

- https://intel471.com/blog/privateloader-malware
- https://www.zscaler.com/blogs/security-research/peeking-privateloader
- https://www.bitsight.com/blog/tracking-privateloader-malware-distribution-service

Chat with our team!

Would you like to know more about our solutions?
Do you want to discover our XDR and CTI products?
Do you have a cybersecurity project in your organization?
Make an appointment and meet us!

Read other contents :

CTI Cybercrime Dark Web

Share this post:
